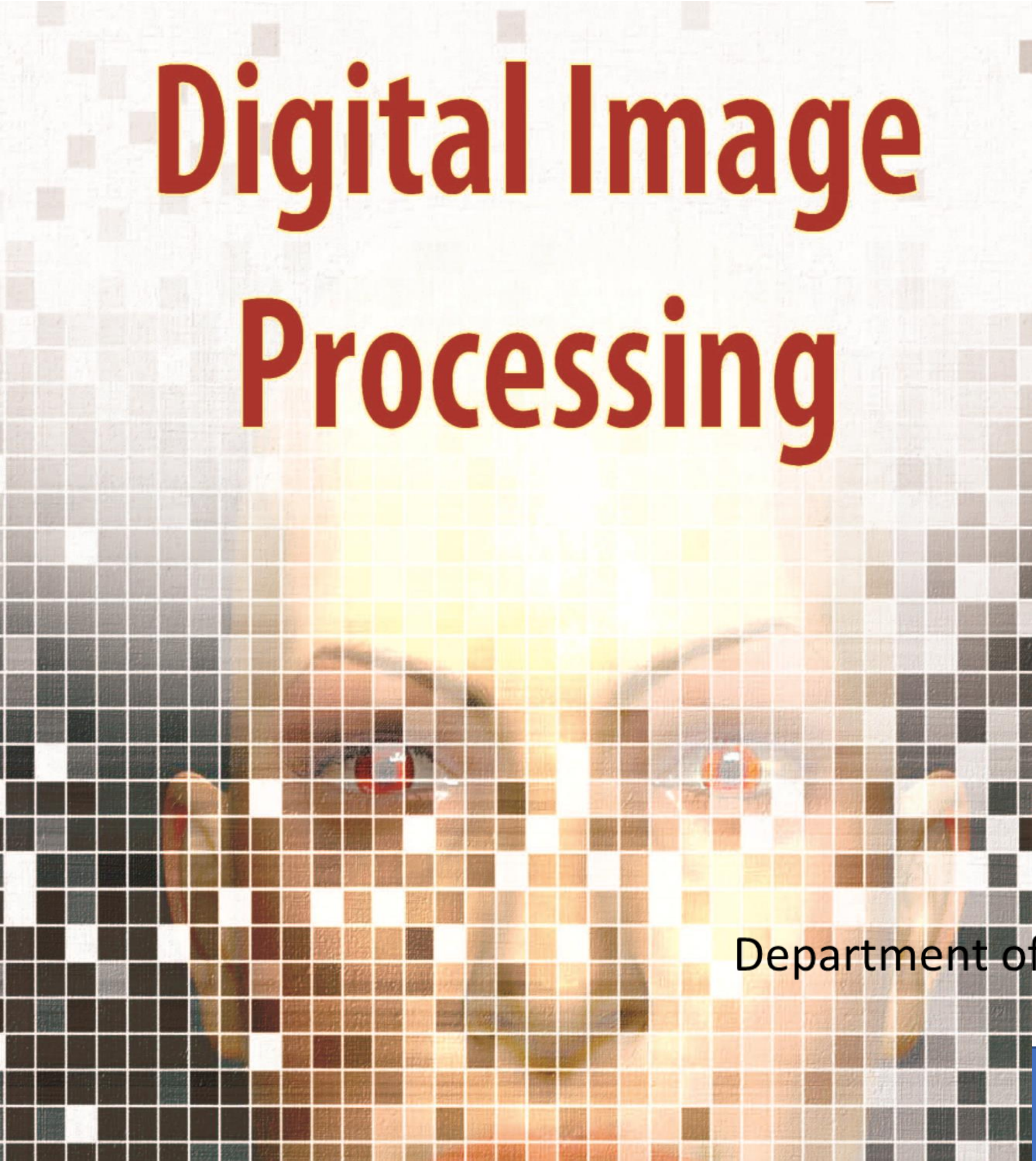
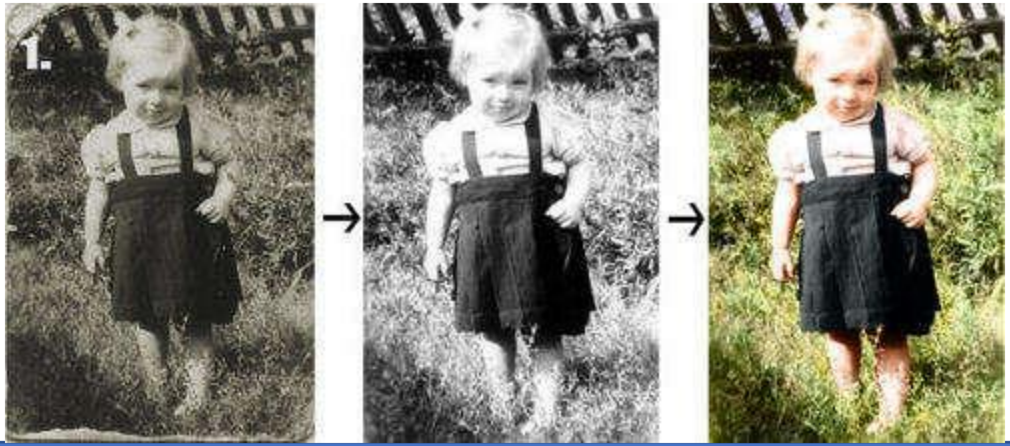


Image Restoration

Prof. Sheli Sinha Chaudhuri
Department of Electronics and Telecommunication Engineering
Jadavpur University

Chapter 5
Image Restoration



Contents

A Model of the Image Degradation/Restoration Process

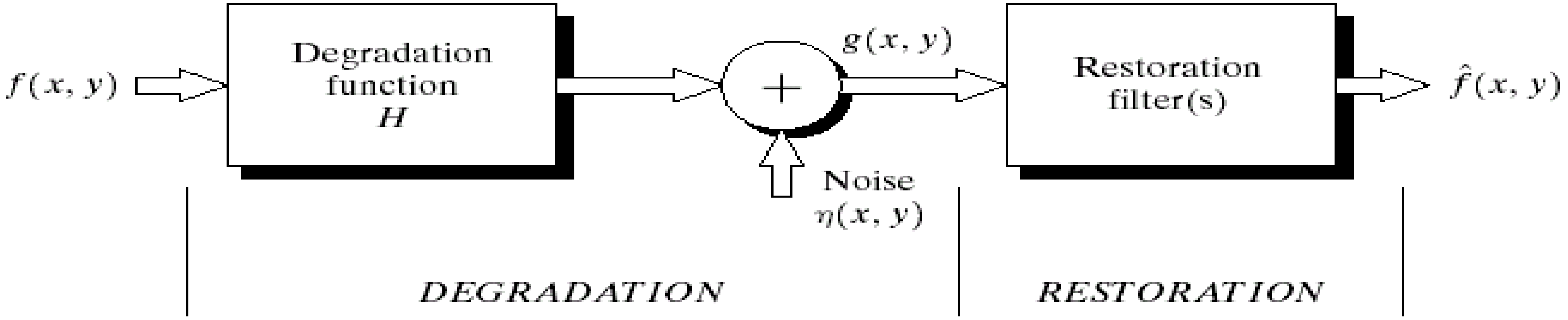


FIGURE 5.1 A model of the image degradation/ restoration process.

A Model of the Image Degradation/Restoration Process

- Degradation
  - Degradation function H
  - Additive noise
  - Spatial domain

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$$

  - Frequency domain

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

Restoration Model

- Restoration

$$g(x, y) \Rightarrow \text{Restoration Filter} \Rightarrow \hat{f}(x, y)$$

Noise Models

- Sources of noise
  - Image acquisition, digitization, transmission
- White noise
  - The Fourier spectrum of noise is constant
- Assuming
  - Noise is independent of spatial coordinates
  - Noise is uncorrelated with respect to the image itself

Gaussian Noise

- Gaussian noise
  - The PDF of a Gaussian random variable, z,

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$

  - Mean:µ
  - Standard deviation: σ
  - Variance: σ^2

- 70% of its values will be in the range

$$[(\mu-\sigma),(\mu+\sigma)]$$

- 95% of its values will be in the range

$$[(\mu-2\sigma),(\mu+2\sigma)]$$

Rayleigh Noise Model

- Rayleigh noise
  - The PDF of Rayleigh noise,

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases}$$

  - Mean: $\mu = a + \sqrt{\pi b/4}$
  - Variance: $\sigma^2 = \dfrac{b(4-\pi)}{4}$

Erlang (Gamma) Noise

- Erlang (Gamma) noise
  - The PDF of Erlang noise, $a >, 0$ is a positive integer, b

$$p(z) = \begin{cases} \dfrac{a^b z^{b-1}}{(b-1)!} e^{-a z} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases}$$

  - Mean: $\mu = \dfrac{b}{a}$
  - Variance: $\sigma^2 = \dfrac{b}{a^2}$

Salt and Pepper Noise

- Impulse (salt-and-pepper) noise
  - The PDF of (bipolar) impulse noise,

$$p(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$$

  - $b > a$: gray-level b will appear as a light dot, while level a will appear like a dark dot
  - Unipolar: either P_a or P_b is zero
- Usually, for an 8-bit image, a=0 (black) and b=0 (white)

Noise Modelling

- Modeling
  - Gaussian
    - Electronic circuit noise, sensor noise due to poor illumination and/or high temperature
  - Rayleigh
    - Range imaging
  - Exponential and gamma
    - Laser imaging
  - Impulse
    - Quick transients, such as faulty switching
  - Uniform
    - Least descriptive
    - Basis for numerous random number generators

Test Pattern for observing Effect of Noise Models

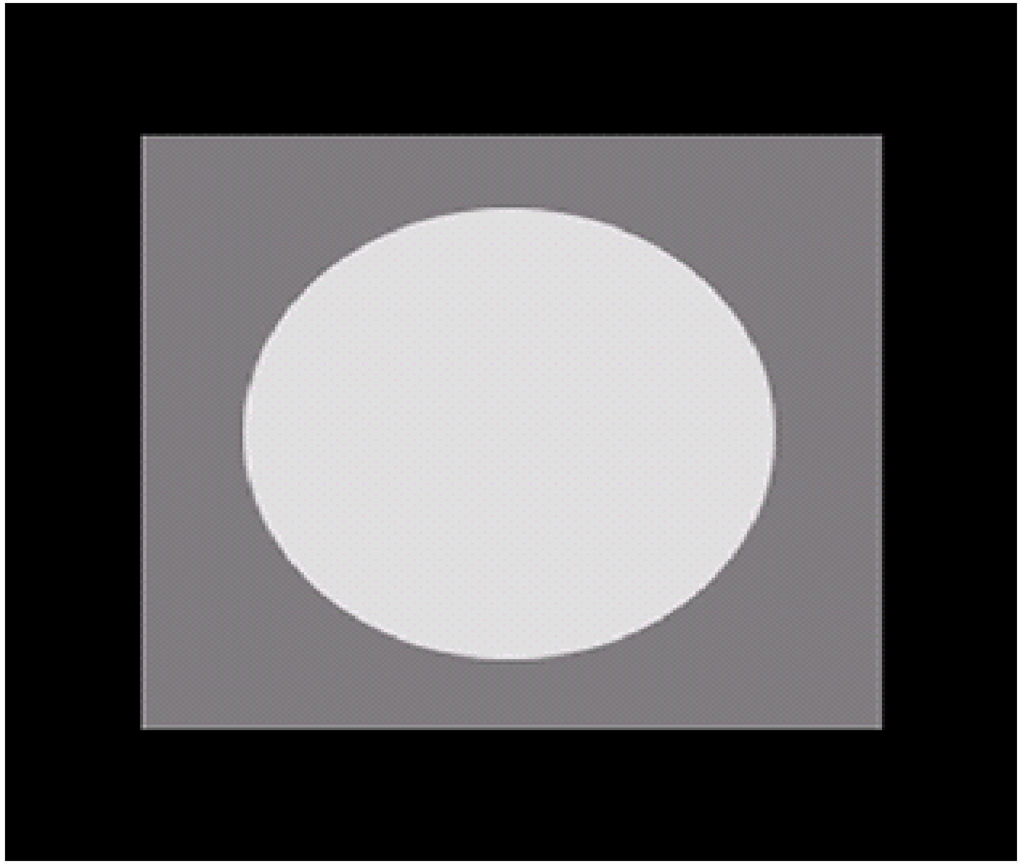


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

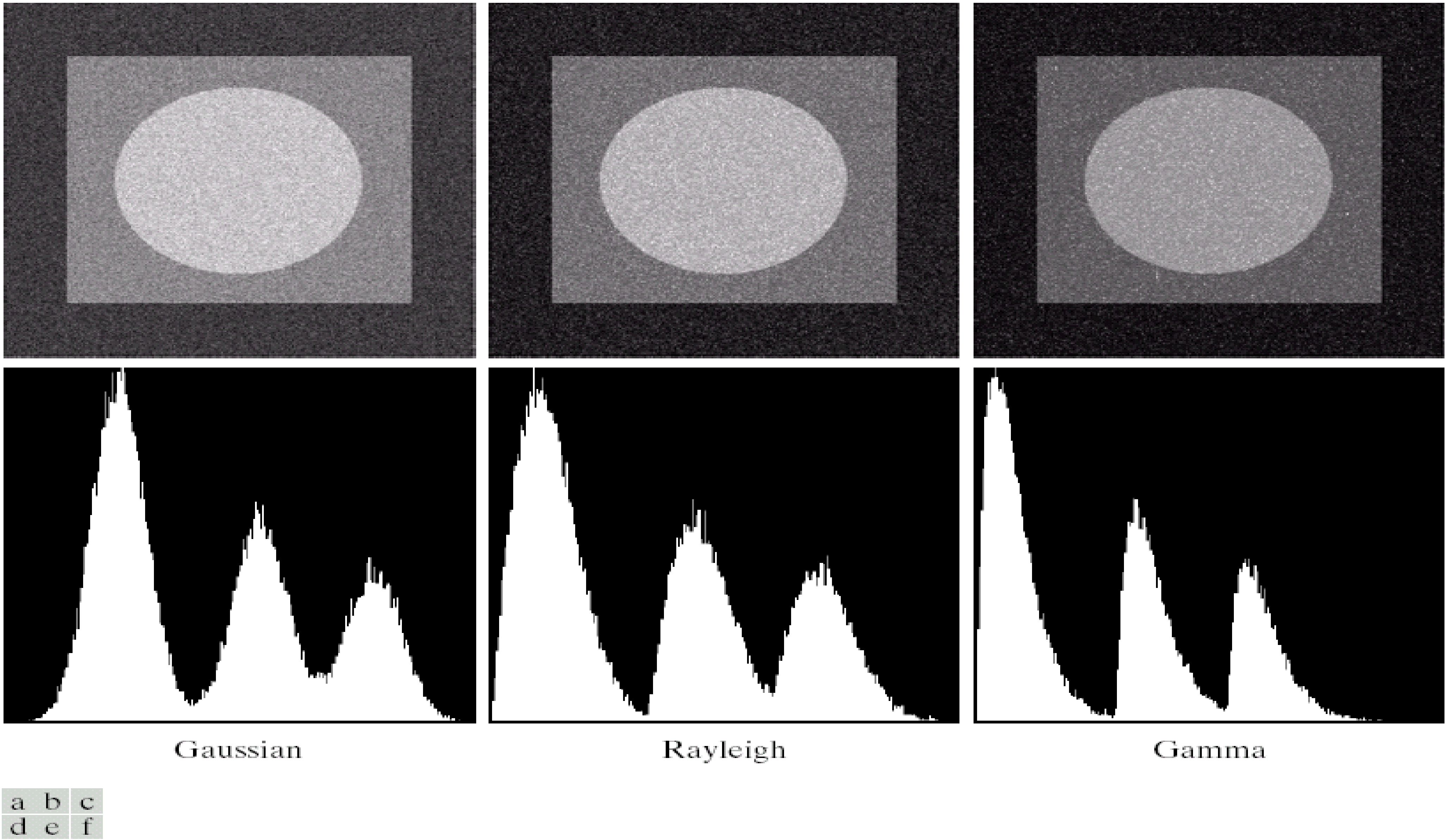


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

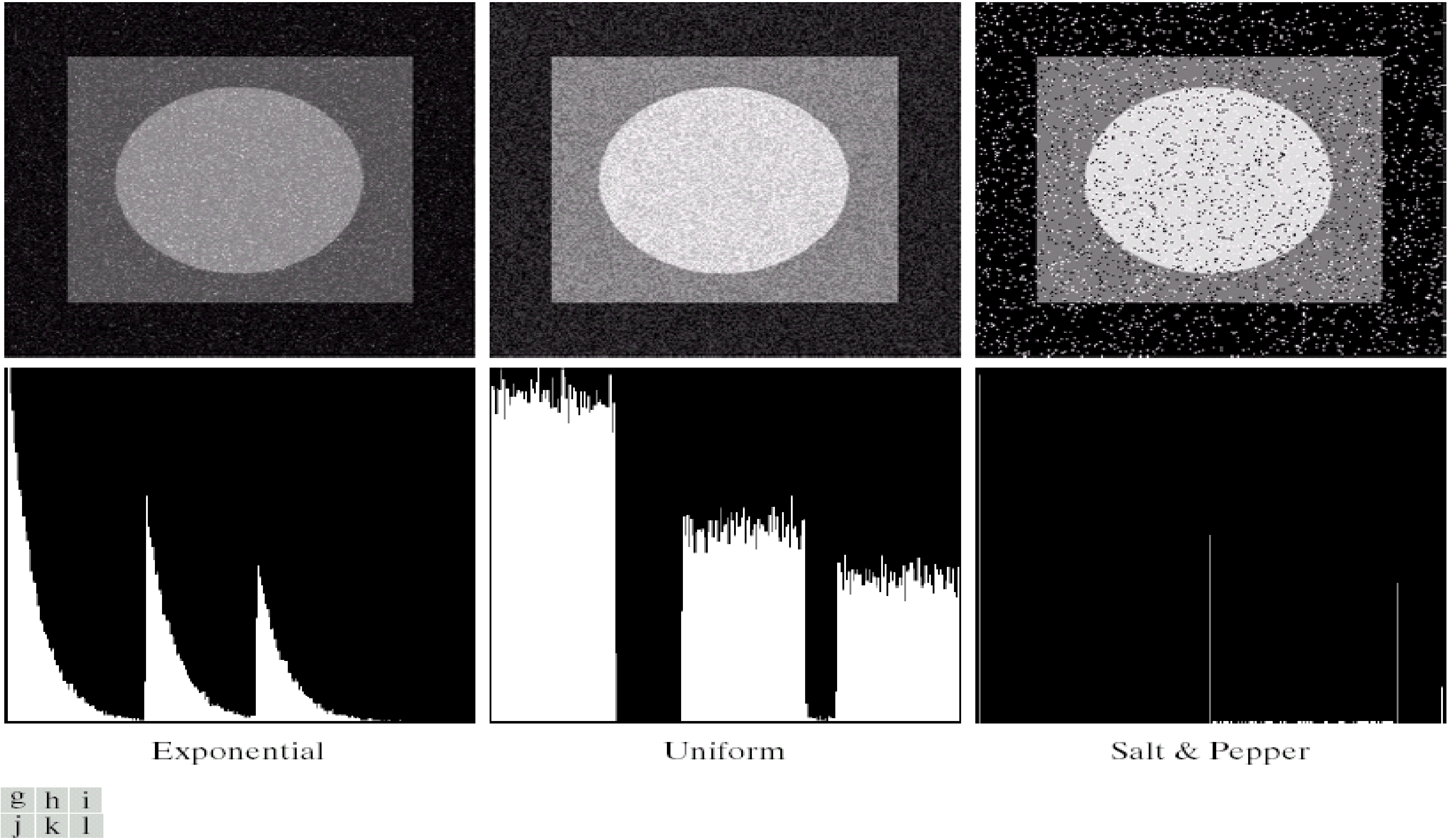


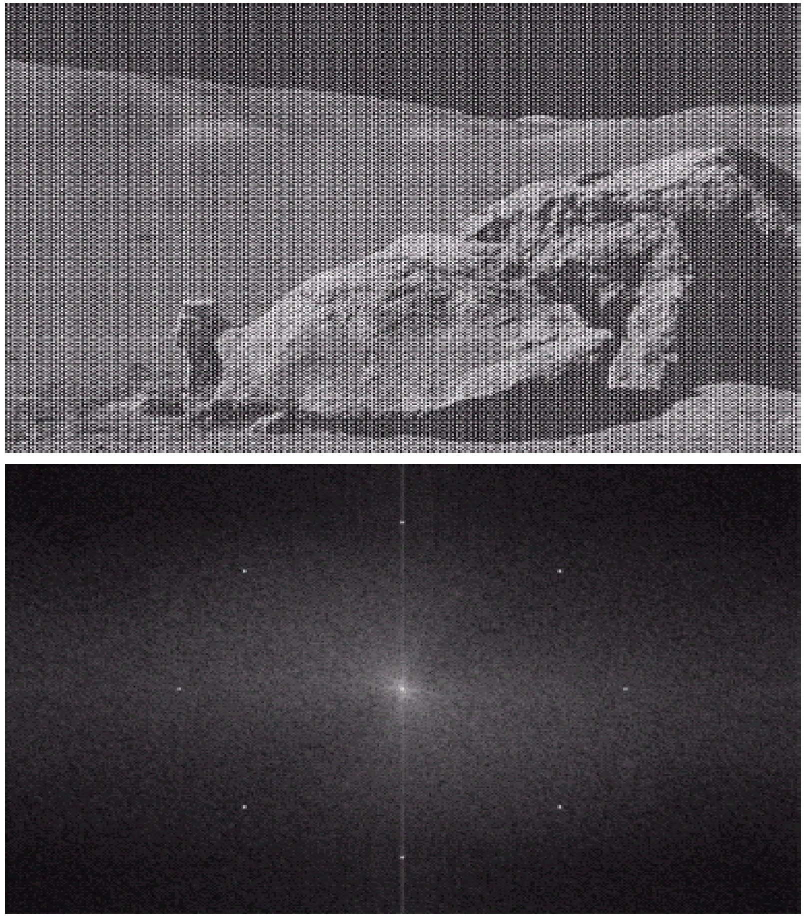
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

Periodic Noise

- Periodic noise
  - Arises typically from electrical or electromechanical interference
  - Reduced significantly via frequency domain filtering

a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)



Noise Parameter Estimation

- Estimation of noise parameters
  - Inspection of the Fourier spectrum
  - Small patches of reasonably constant gray level
    - For example, 150*20 vertical strips
    - Calculate μ, σ, a , b from

$$\mu = \sum_{z_i \in S} z_i p(z_i)$$

$$\sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 p(z_i)$$

Histograms of Noisy Images

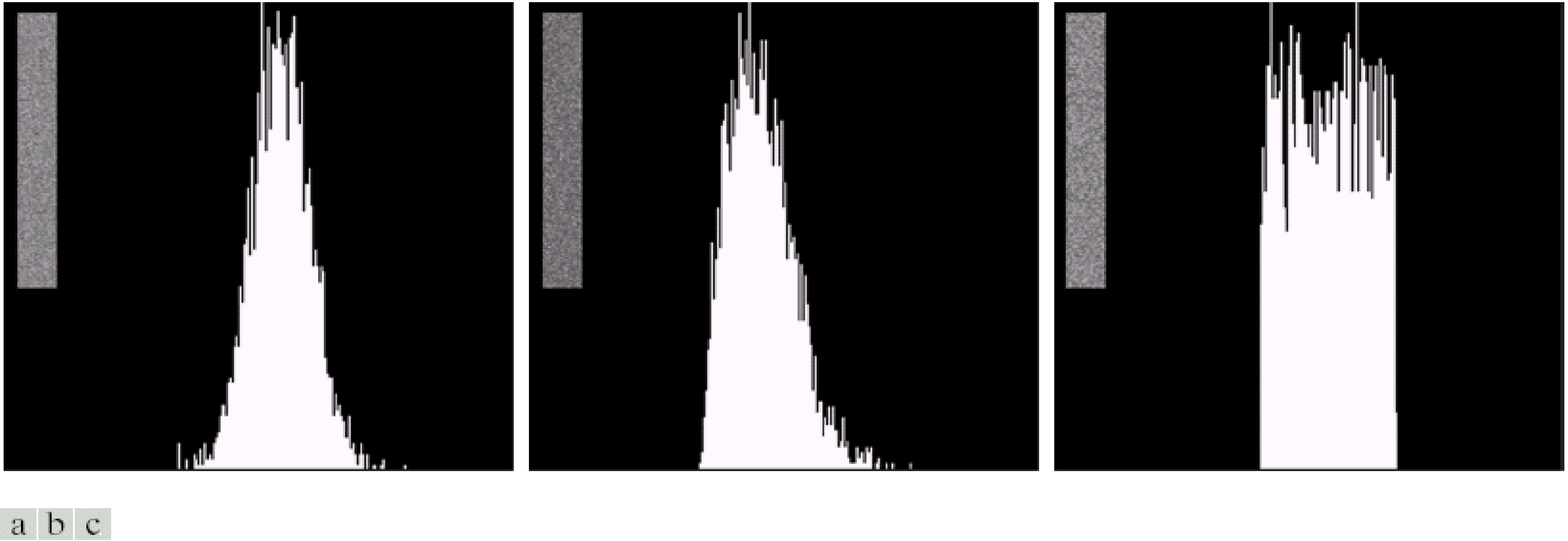


FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

Restoration in the Presence of Noise Only-Spatial Filtering

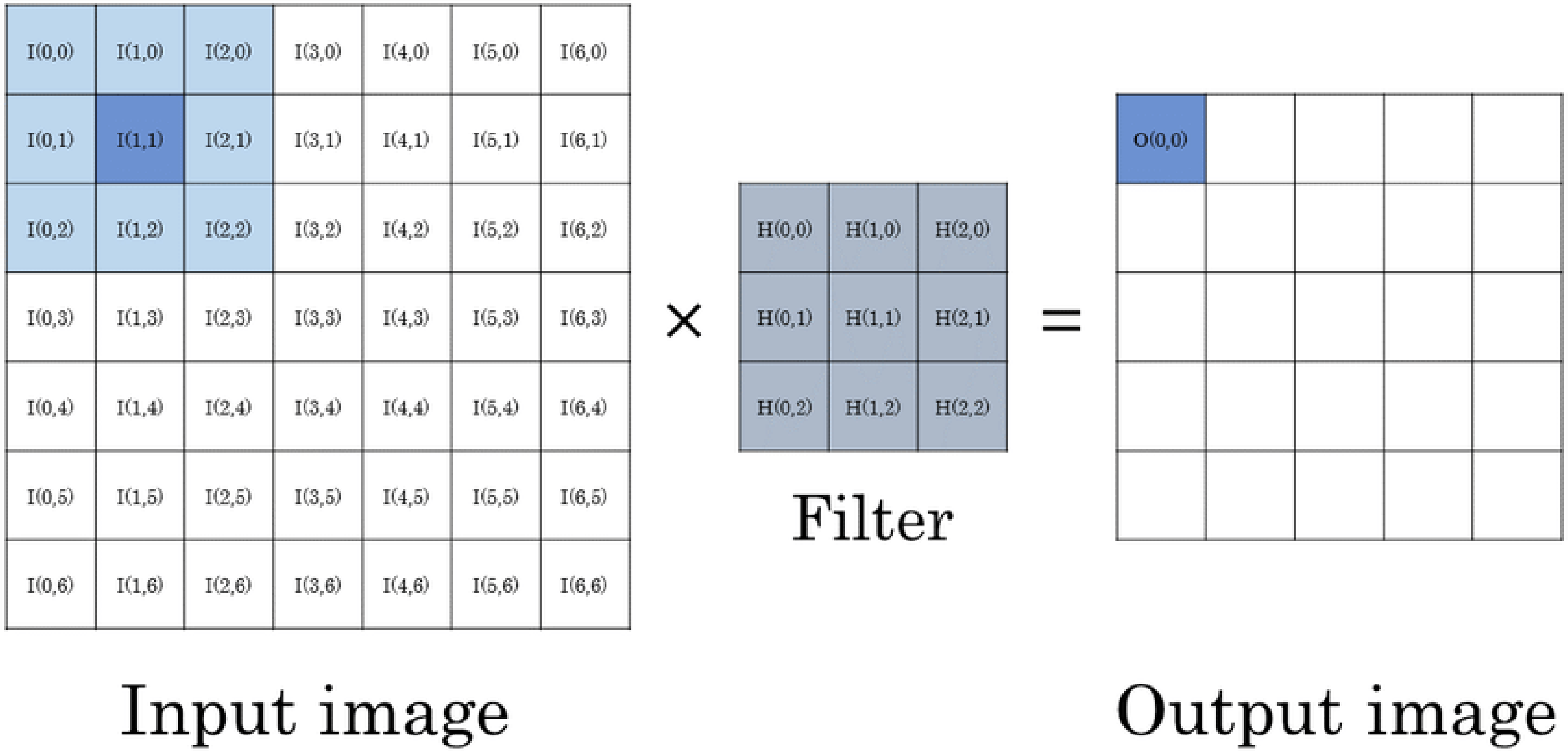
- Degradation
  - Spatial domain

$$g(x, y) = f(x, y) + \eta(x, y)$$

  - Frequency domain

$$G(u, v) = F(u, v) + N(u, v)$$

Sliding Window example



Sliding Window example

7	23	50	64	14
15	13	31	46	8
42	25	92	31	32
71	44	74	94	92
2	43	51	35	4

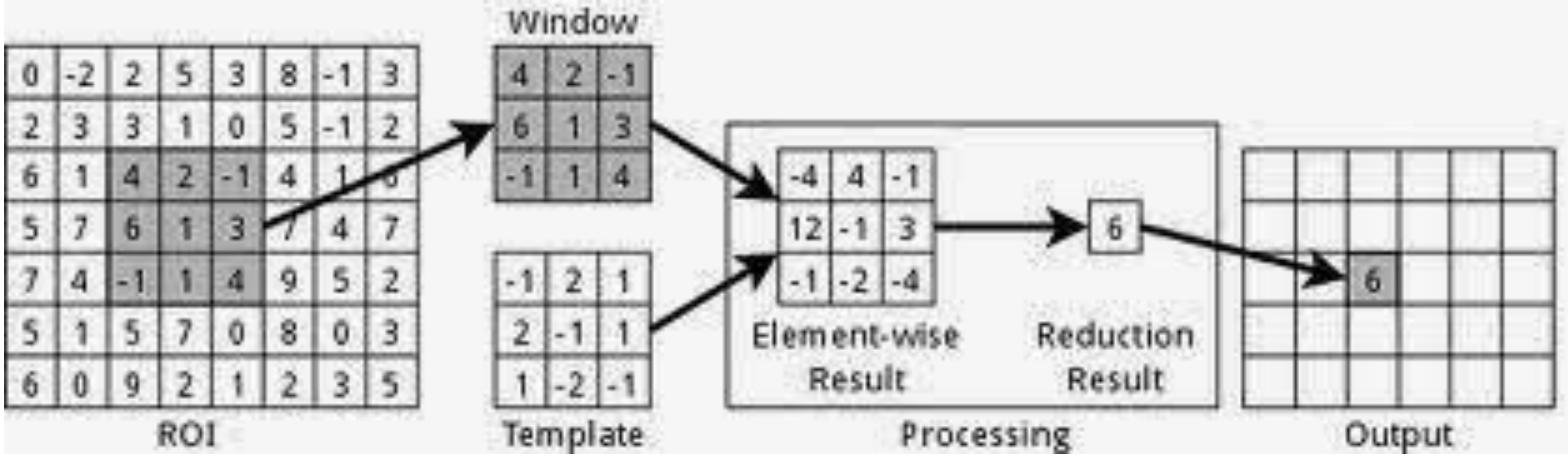
X

0	2	0
0	0	0
0	0	0

=

-	-	-	-	-
-	46	100	128	-
-	26	62	92	-
-	50	184	62	-
-	-	-	-	-

Sliding Window example



Mean Filter

- Mean filters
  - Arithmetic mean filter

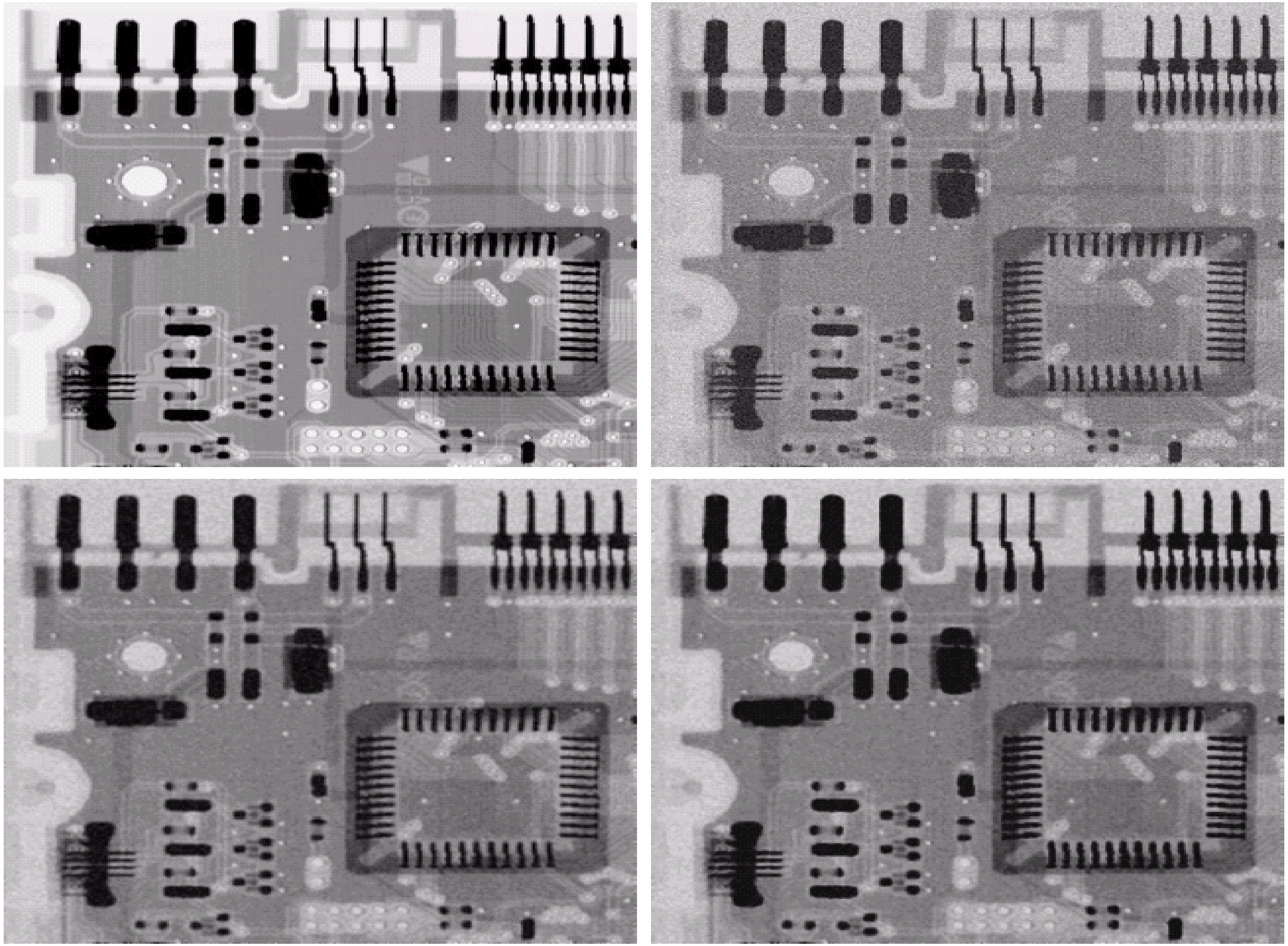
$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

  - Geometric mean filter

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

Mean Filter

- Usage
  - Arithmetic and geometric mean filters: suited for Gaussian or uniform noise
  - Harmonic and Contraharmonic filters: suited for impulse noise



a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Harmonic Mean Filter

- Harmonic mean filter
  - Works well for salt noise, but fails for pepper noise

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

Contraharmonic Mean Filter

- Contraharmonic mean filter
  - $Q > 0$: eliminates pepper noise
  - $Q < 0$: eliminates salt noise

$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Q is the order of the filter.

This filter becomes the arithmetic mean filter if Q=0 and the harmonic mean filter if Q=-1.

a	b
c	d

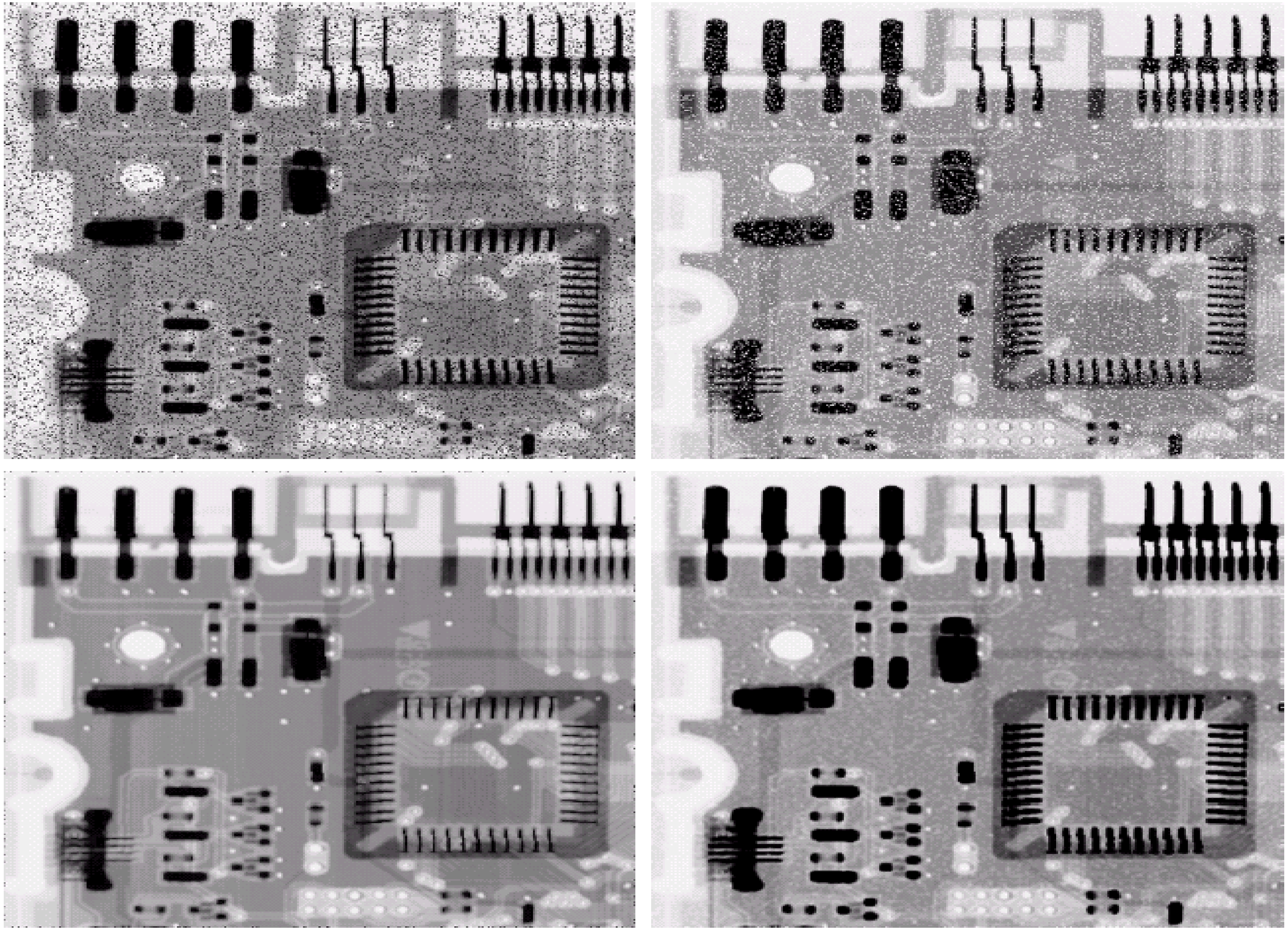
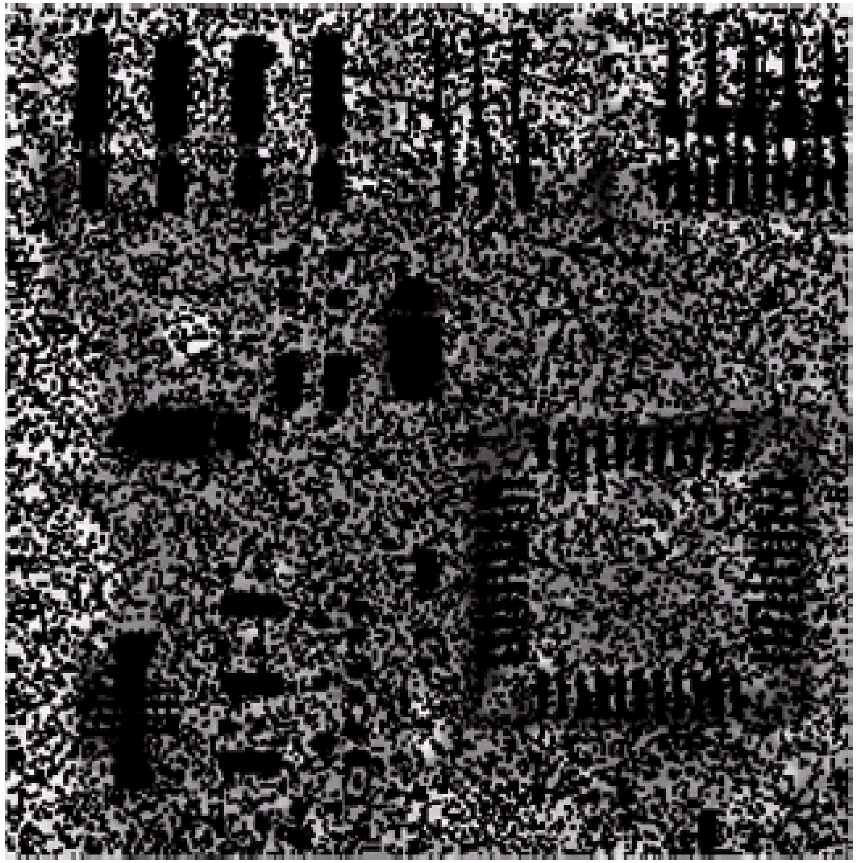
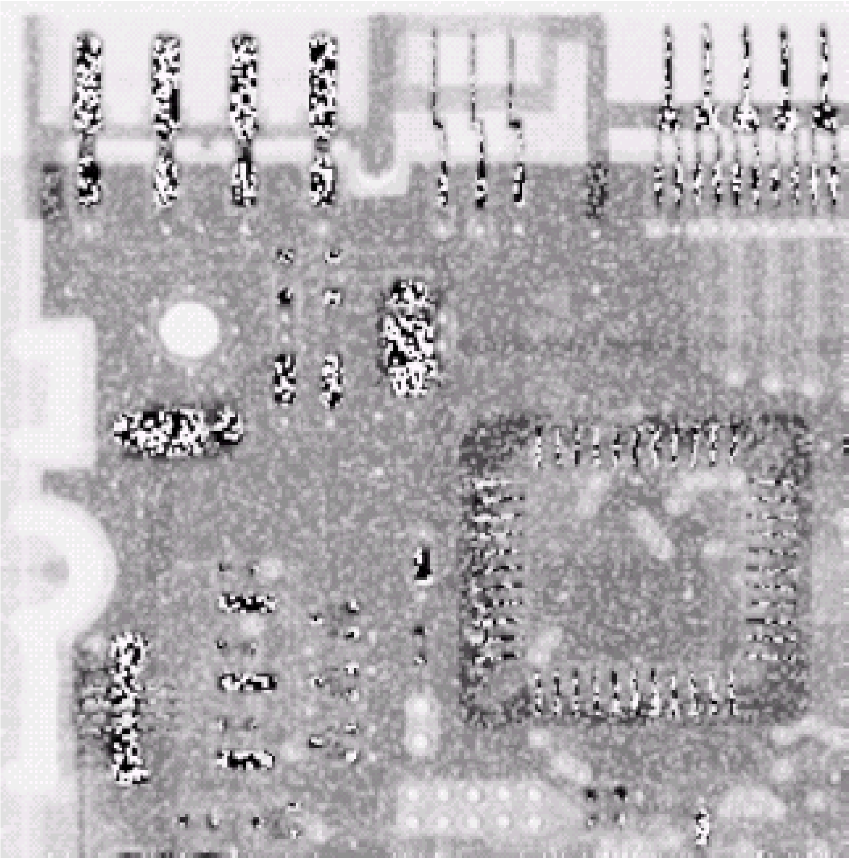


FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

Selection of Parameters for Contraharmonic Filter





a | b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

Median Filter

- Order-statistics filters
  - Median filter
    - Effective in the presence of both bipolar and unipolar impulse noise

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s,t)\}$$

Max and Min Filters

- Max and min filters
  - max filters reduce pepper noise

$$\hat{f}(x, y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

  - min filters reduce salt noise

$$\hat{f}(x, y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Midpoint filter

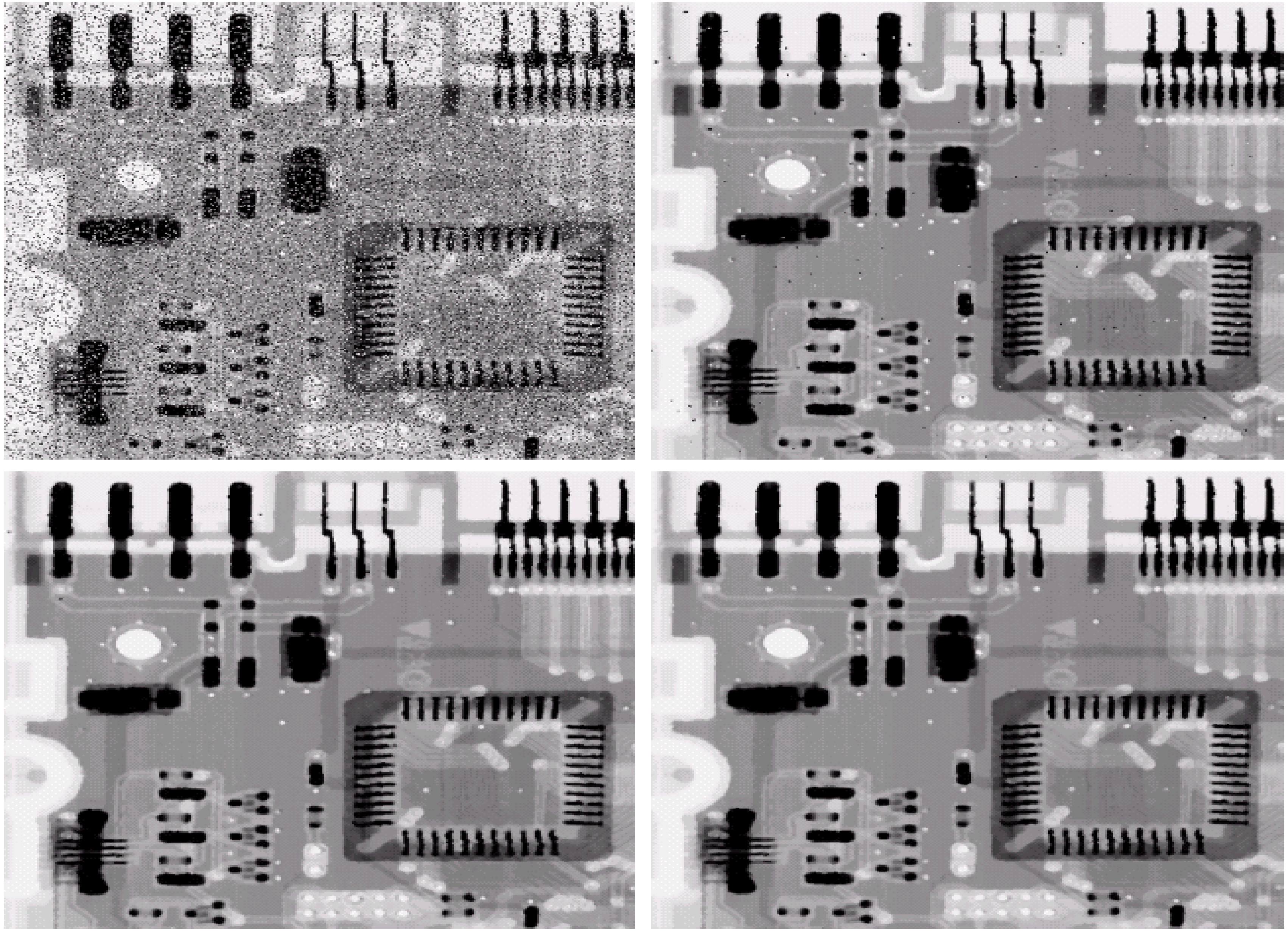
- Midpoint filter
  - Works best for randomly distributed noise, like Gaussian or uniform noise

$$\hat{f}(x,y)=\frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\}+\min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

Alpha-trimmed mean filter

- Alpha-trimmed mean filter
  - Delete the d/2 lowest and the d/2 highest gray-level values
  - Useful in situations involving multiple types of noise, such as a combination of salt-and-pepper and Gaussian noise

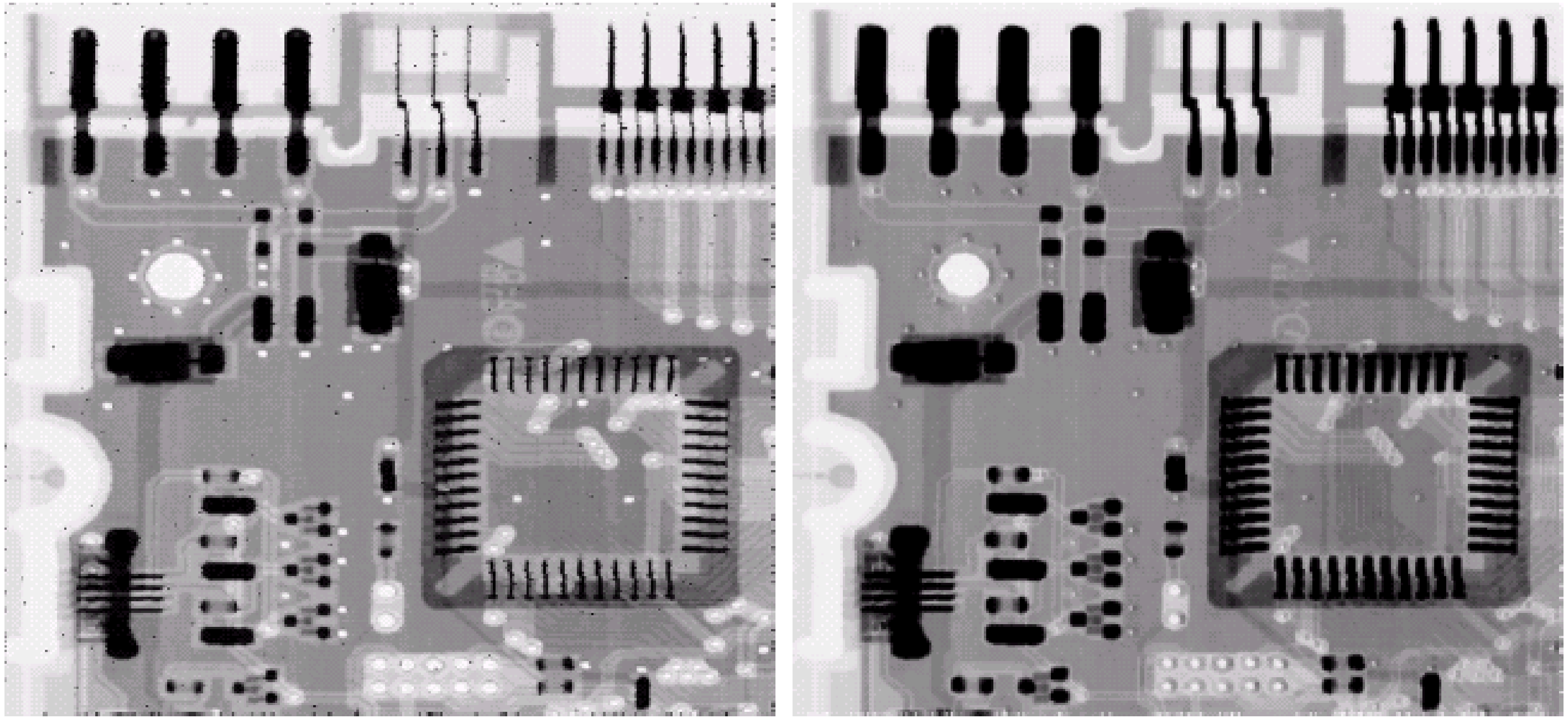
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$



a	b
c	d

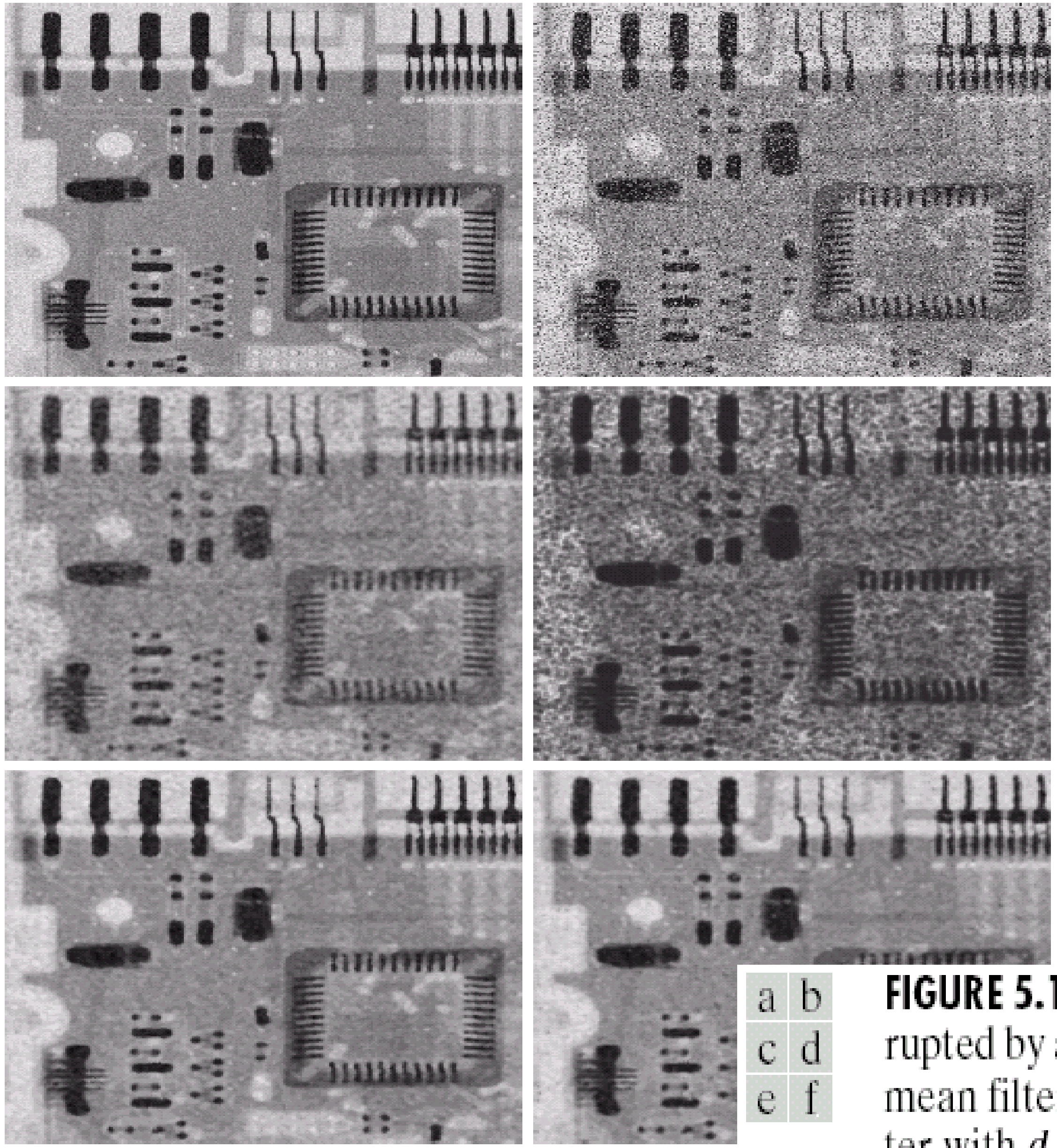
FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Max and Min Filter



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.



a	b
c	d
e	f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

Adaptive Filter

- Adaptive, local noise reduction filter whose behavior changes based on the statistical characteristics of the image inside the filter region.

  - If σ_η^2 is zero, return simply the value of $g(x, y)$
  - If $\sigma_\eta^2 < \sigma_L^2$, return a value close to $g(x, y)$
  - If $\sigma_\eta^2 = \sigma_L^2$, return the arithmetic mean value m_L

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(x, y) - m_L\right]$$

a	b
c	d

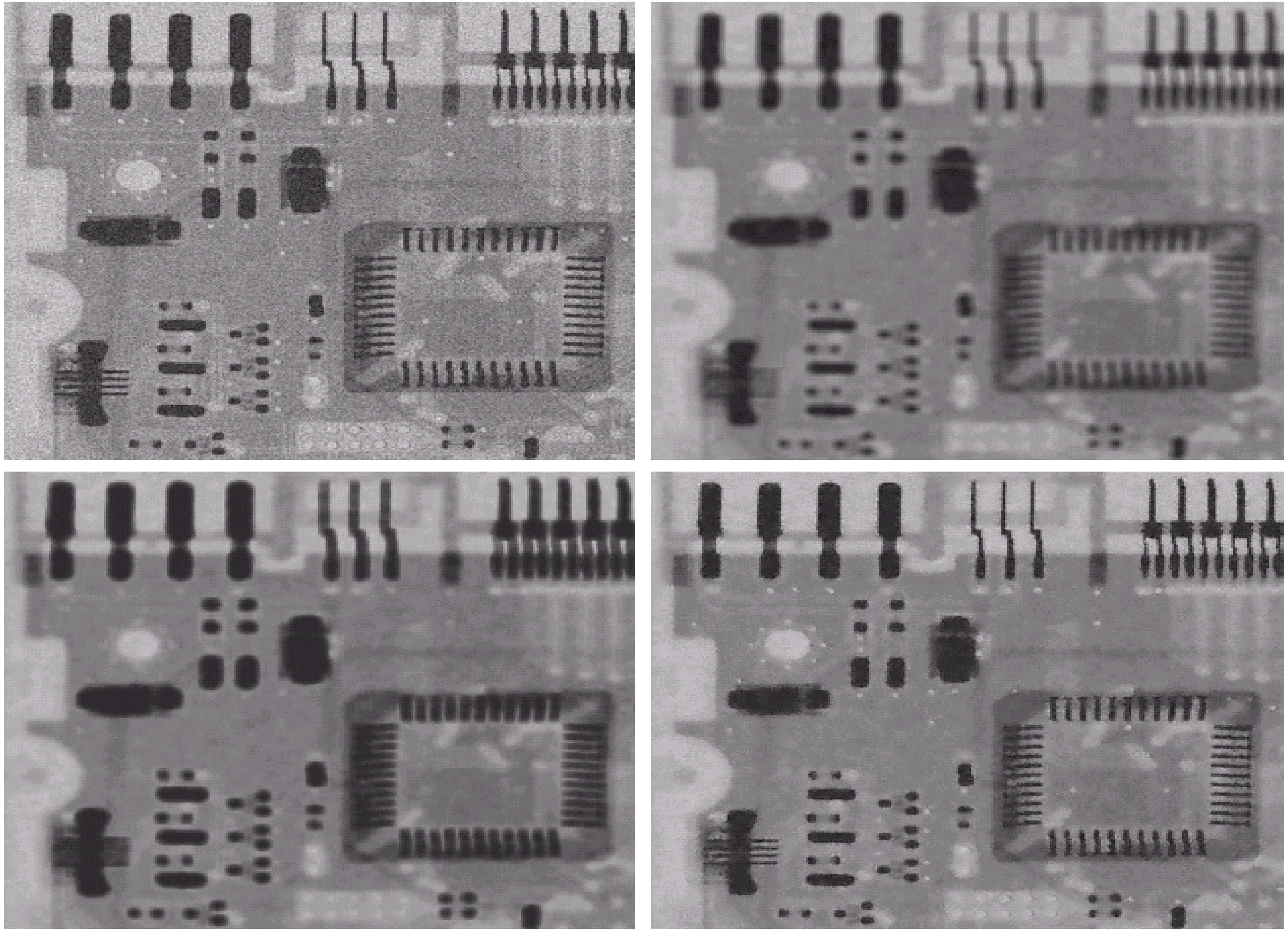


FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7×7.

Adaptive Median Filter

- Adaptive median filter
  - z_{min} = minimum gray level value in S_{xy}
  - z_{max} = maximum gray level value in S_{xy}
  - z_{med} = median of gray levels in S_{xy}
  - z_{xy} = gray level at coordinates (x, y)
  - S_{max} = maximum allowed size of S_{xy}

Adaptive Median Filter

- Algorithm: Works in two stages
- Level A: A1= $z_{med} - z_{\min}$
- A2= $z_{med} - z_{\max}$
- If A1>0 and A2<0, go to level B
- Else increase the window size
- If window size $\leq S_{\max}$ repeat level A
- Else output z_{med}

- Level B: B1= $z_{xy} - z_{\min}$
- B2= $z_{xy} - z_{\max}$
- If B1>0 AND B2<0, output z_{xy}
- Else output z_{med}

Adaptive Median Filter

- Purposes of the algorithm
  - Remove salt-and-pepper (impulse) noise
  - Provide smoothing
  - Reduce distortion, such as excessive thinning or thickening of object boundaries

Adaptive Median Filter

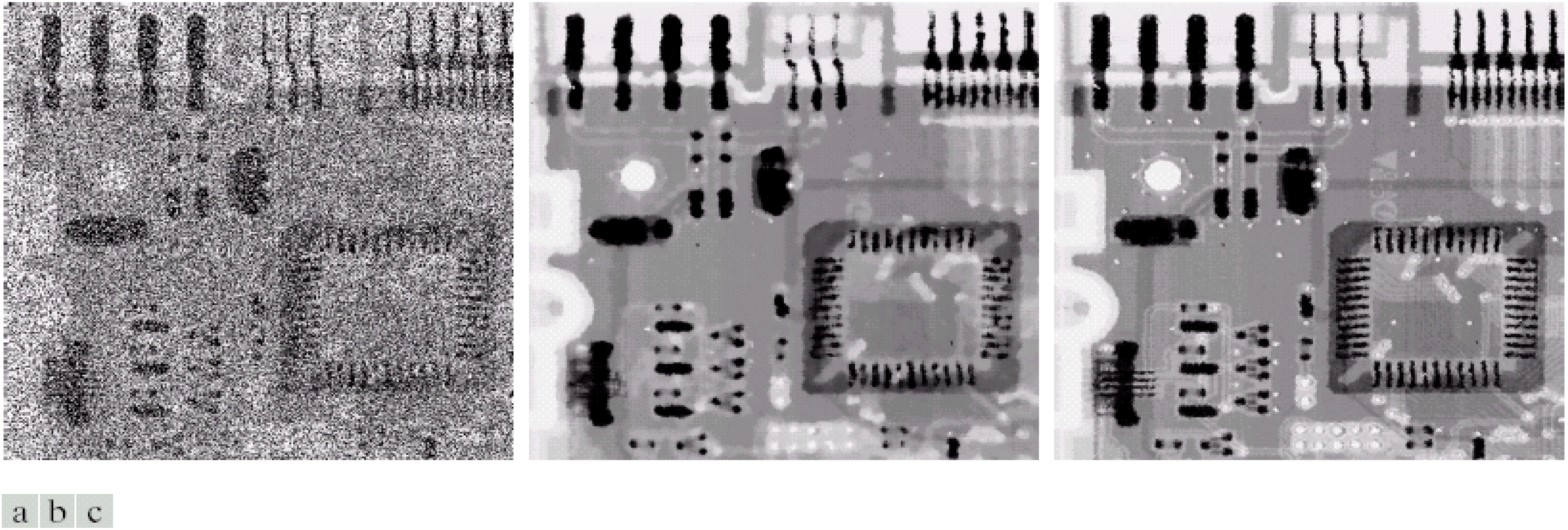


FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7 × 7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

Periodic Noise Reduction by Frequency Domain Filtering

- Bandreject filters
  - Ideal Bandreject filter

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \dfrac{W}{2} \\ 0 & \text{if } D_0 - \dfrac{W}{2} \le D(u,v) \le D_0 + \dfrac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \dfrac{W}{2} \end{cases}$$

$$D(u,v) = \left[(u - M/2)^2 + (v - N/2)^2\right]^{1/2}$$

Band Reject Filter

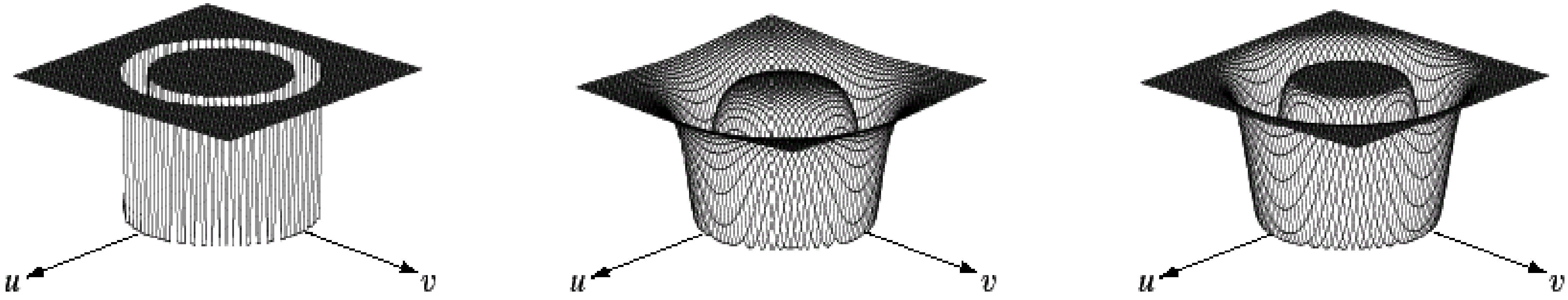
- Butterworth bandreject filter of order n

$$H(u,v) = \frac{1}{1+\left[\frac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}}$$

- Gaussian bandreject filter

$$H(u,v) = 1 - e^{-\frac{1}{2}\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^2}$$

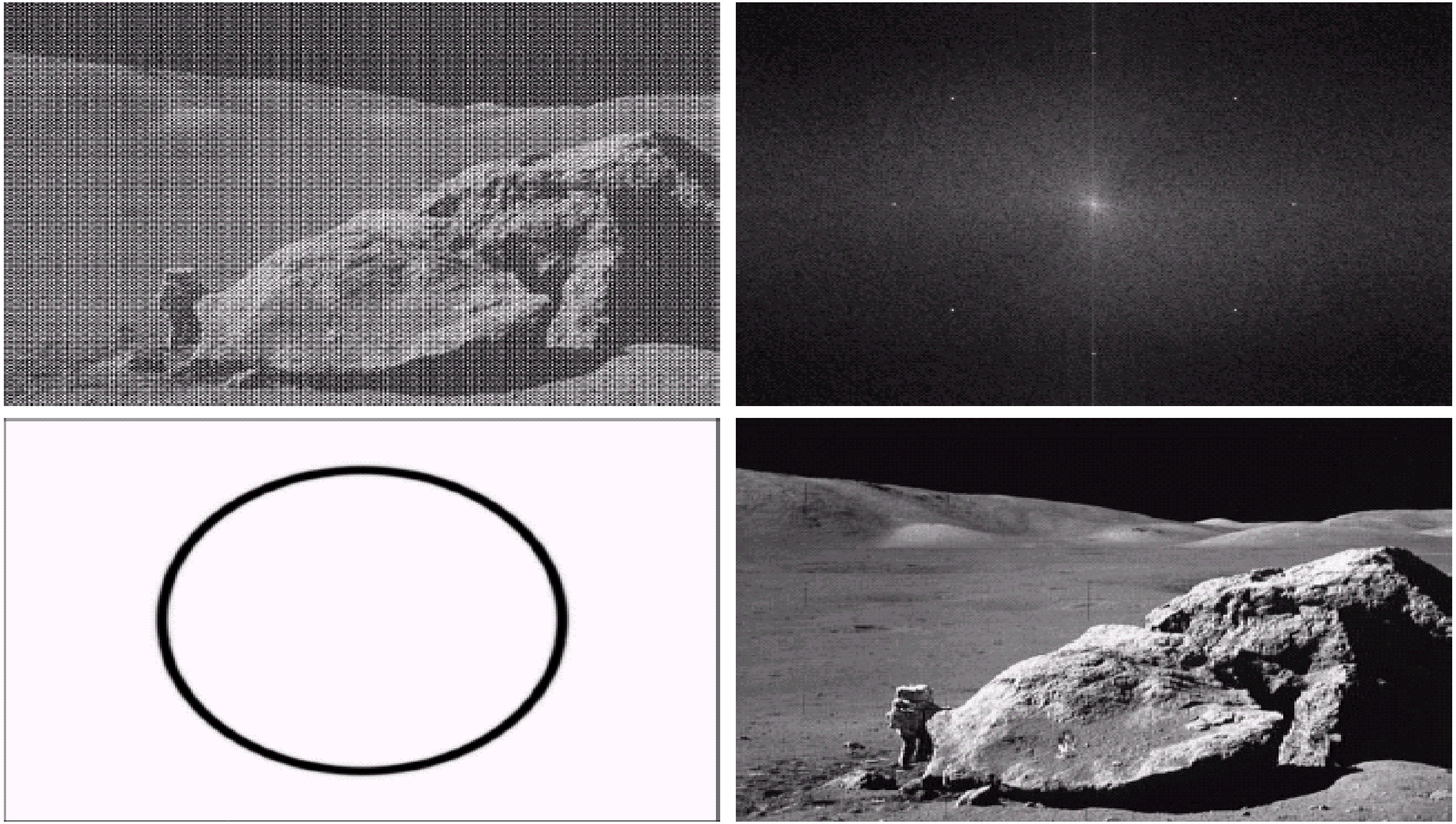
Band Reject Filter



a b c

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

Band Reject Filter



a	b
c	d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

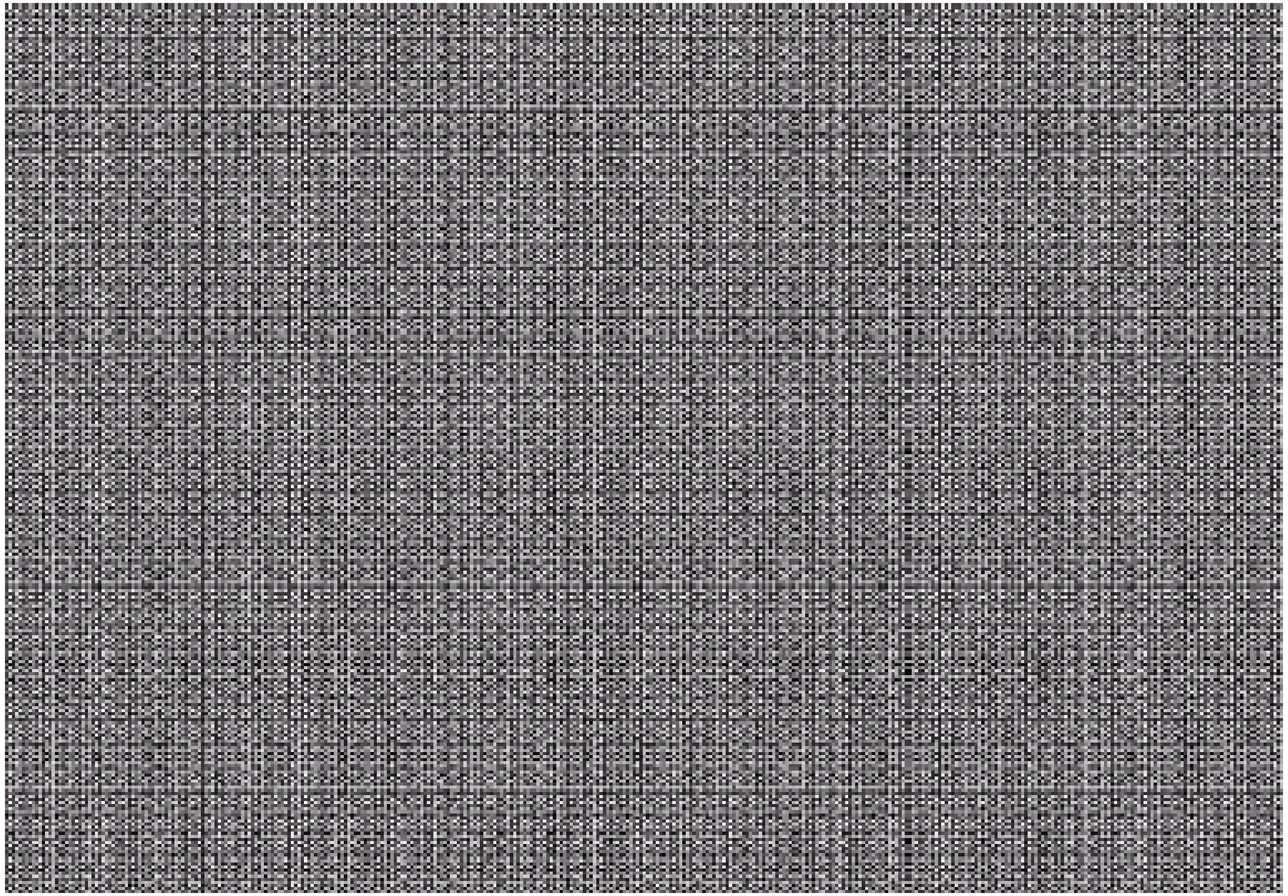
Band Pass Filters

- Bandpass filters

$$H_{bp}(u,v) = 1 - H_{br}(u,v)$$

Band Pass Filters

FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



Notch Filters

- Notch filters- Filters that reject or pass frequencies in pre-defined neighbourhoods about a centre frequency.
    - Ideal notch reject filter

$$H(u,v)=\begin{cases}0 & \text{if } D_1(u,v)\le D_0 \text{ or } D_2(u,v)\le D_0\\ 1 & \text{otherwise}\end{cases}$$

$$D_1(u,v)=\left[(u-M/2-u_0)^2+(v-N/2-v_0)^2\right]^{1/2}$$

$$D_2(u,v)=\left[(u-M/2+u_0)^2+(v-N/2+v_0)^2\right]^{1/2}$$

Notch Filters

- Butterworth notch reject filter of order n

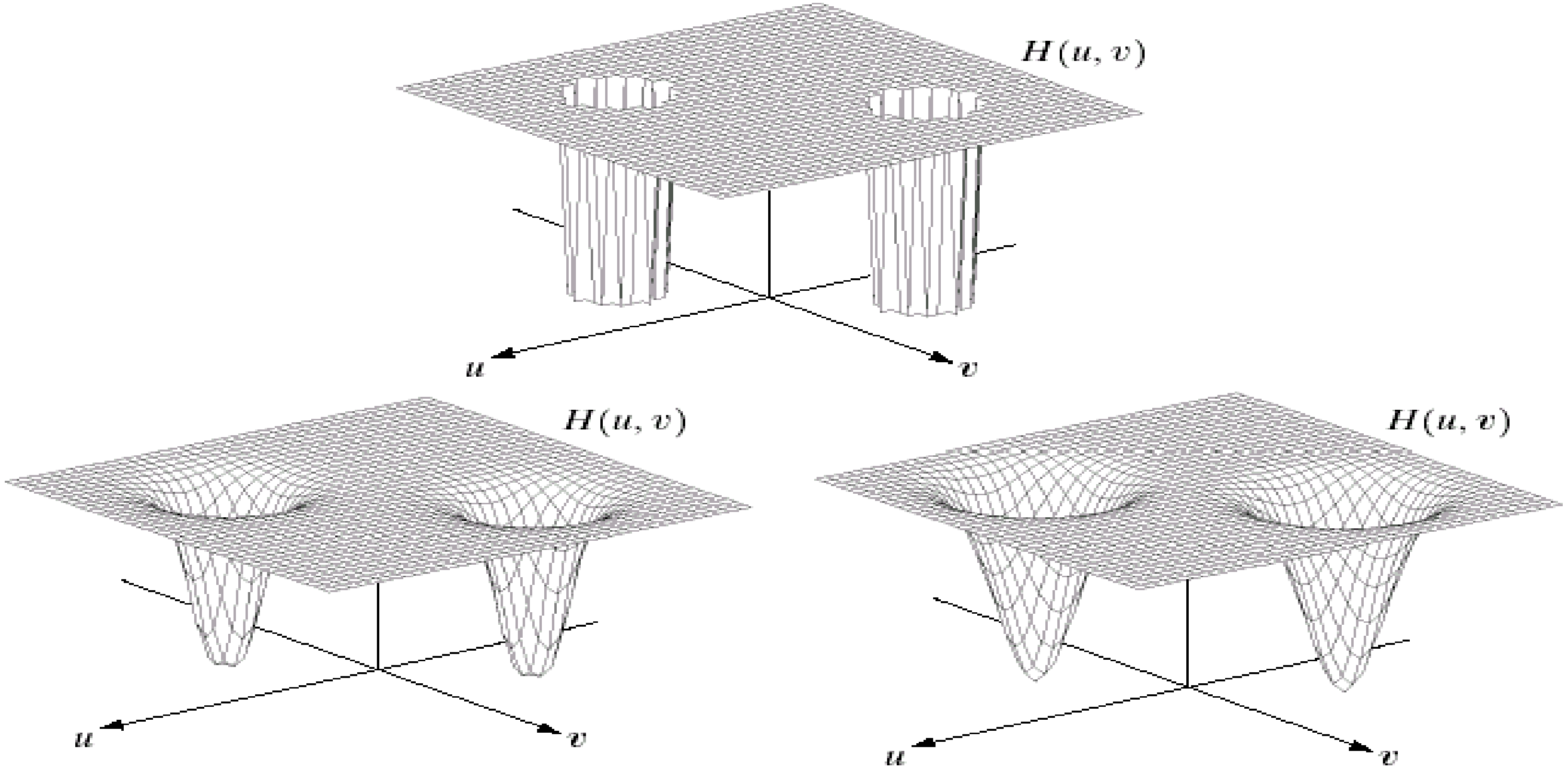
$$H(u,v) = \frac{1}{1 + \left[\frac{D_0^2}{D_1(u,v) D_2(u,v)} \right]^n}$$

Notch Filters

- Gaussian notch reject filter

$$H(u,v)=1-e^{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]}$$

Different Notch Filters



a
b c

FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Notch Filters

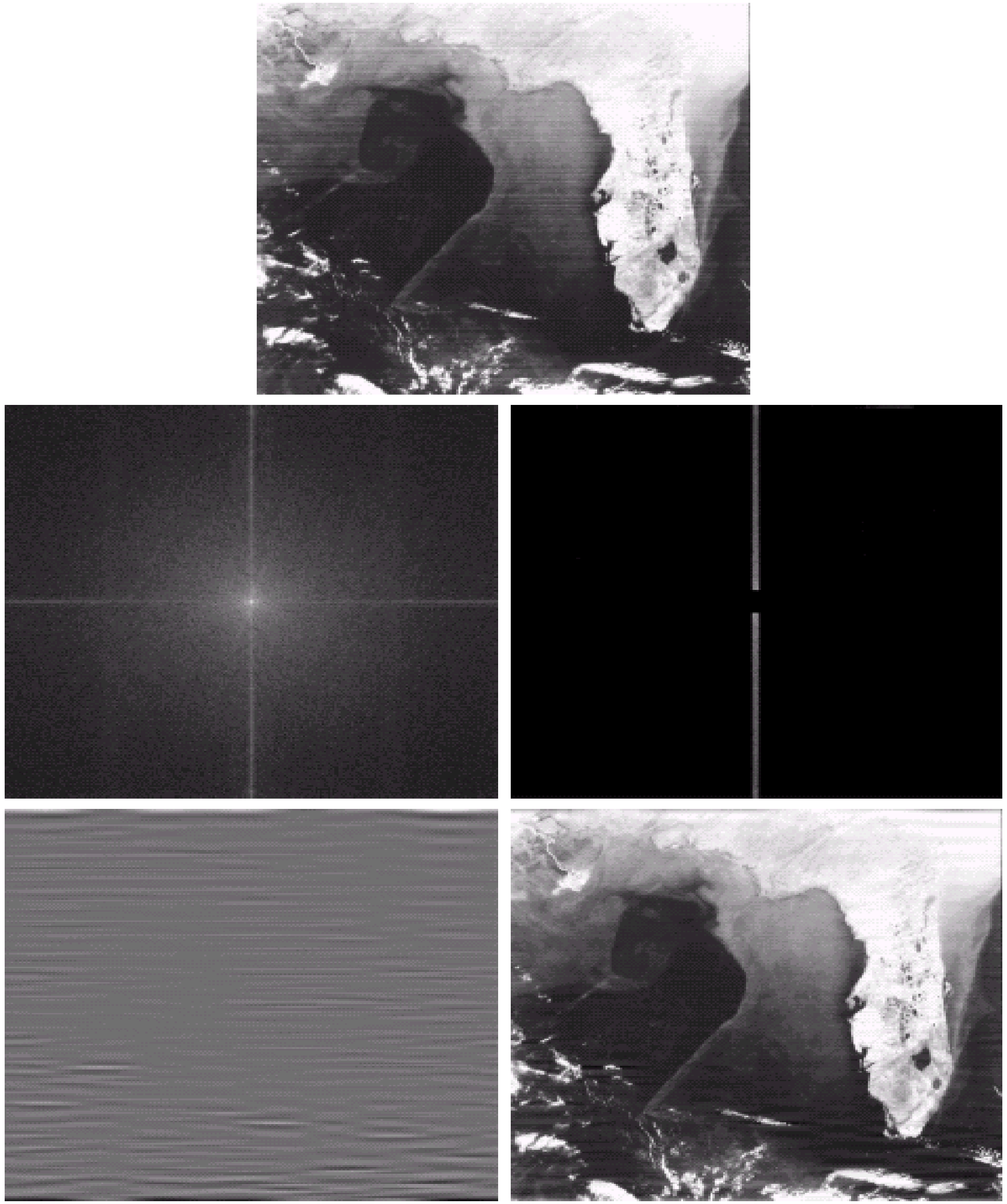
- Notch pass filter

$$H_{np}(u,v) = 1 - H_{nr}(u,v)$$

Notch Filters

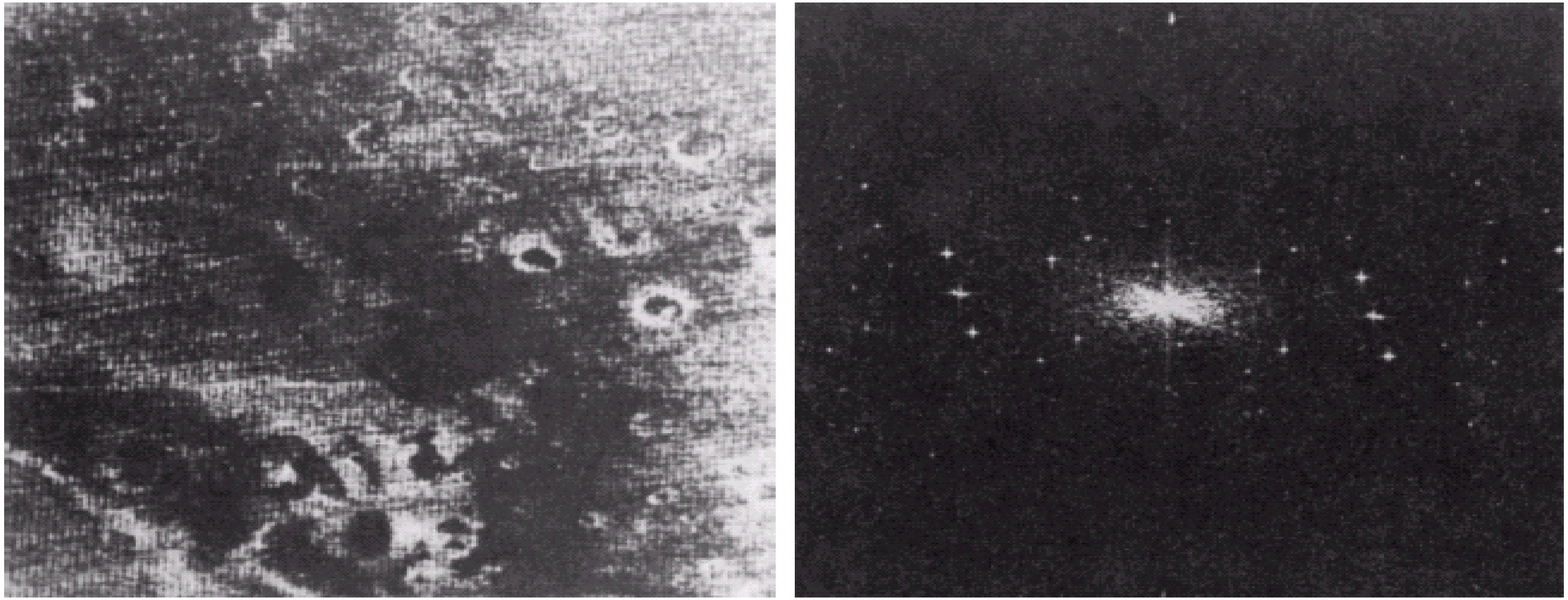
a
b c
d e

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)



Notch Filters

- Optimum notch filtering



a b

FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6.* (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

Interference in Noise Pattern

- Interference noise pattern

$$N(u,v) = H(u,v)G(u,v)$$

- Interference noise pattern in the spatial domain

$$\eta(x,y) = \Im^{-1}\{H(u,v)G(u,v)\}$$

- Subtract from $g(x,y)$ a weighted portion of $f(x,y)$ to obtain an estimate of $\eta(x,y)$

$$\hat{f}(x,y) = g(x,y) - w(x,y)\eta(x,y)$$

Interference in Noise Pattern

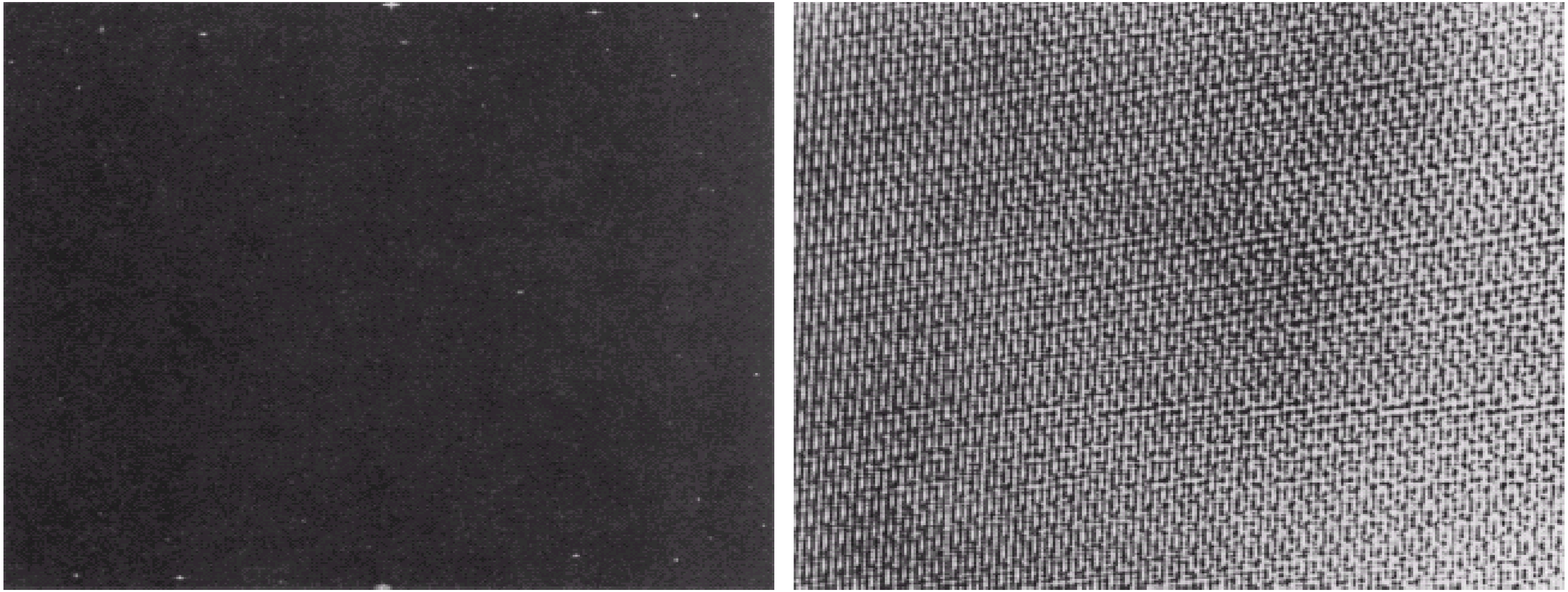
- Minimize the local variance of $\hat{f}(x, y)$
- The detailed steps are listed in Page 251 (page 341)
- Result

$$w(x, y) = \frac{\overline{g(x, y)\eta(x, y)} - \overline{g}(x, y)\overline{\eta}(x, y)}{\overline{\eta^2}(x, y) - \overline{\eta}^2(x, y)}$$

Interference in Noise Pattern

FIGURE 5.21 Fourier spectrum (without shifting) of the image shown in Fig. 5.20(a). (Courtesy of NASA.)

Interference in Noise Pattern



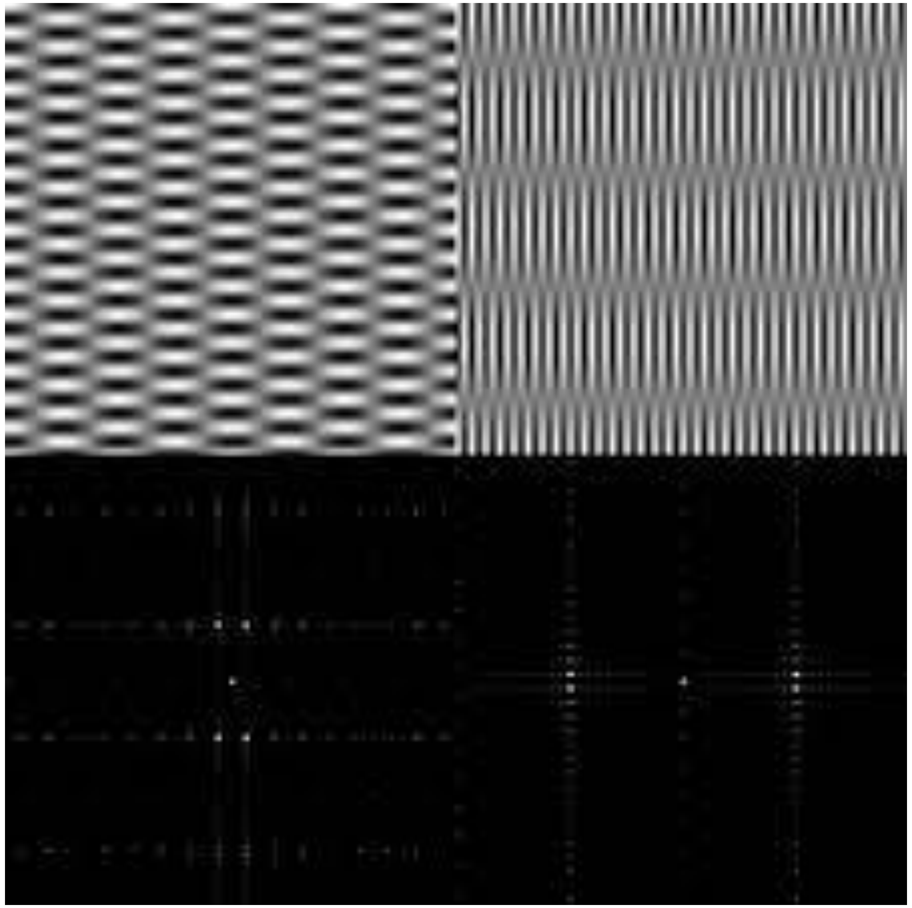
a b

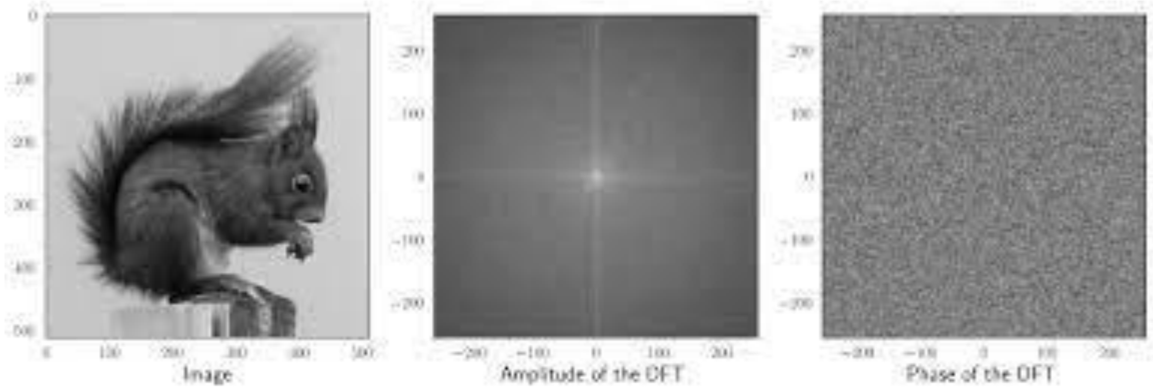
FIGURE 5.22 (a) Fourier spectrum of $N(u, v)$, and (b) corresponding noise interference pattern $\eta(x, y)$. (Courtesy of NASA.)

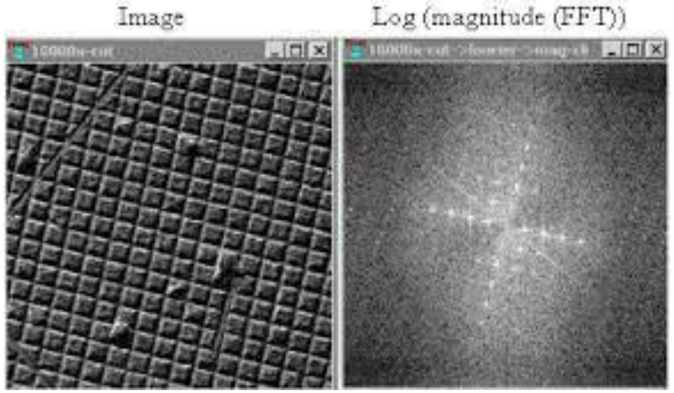
Interference in Noise Pattern

FIGURE 5.23 Processed image. (Courtesy of NASA.)

Example of Fourier Transform







Linear, Position-Invariant Degradations

- Input-output relationship

$$g(x,y) = H[f(x,y)] + \eta(x,y)$$

$$\Downarrow \quad \eta(x,y) = 0$$

$$g(x,y) = H[f(x,y)]$$

Linear, Position-Invariant Degradations

- H is linear if

$$H[af_1(x,y)+bf_2(x,y)] = aH[f_1(x,y)]+bH[f_2(x,y)]$$

  - Additivity

$$H[f_1(x,y)+f_2(x,y)] = H[f_1(x,y)]+H[f_2(x,y)]$$

Linear, Position-Invariant Degradations

- Homogeneity

$$H[af_1(x,y)] = aH[f_1(x,y)]$$

- Position (or space) invariant, if

$$H[f(x-\alpha, y-\beta)] = g(x-\alpha, y-\beta)]$$

for any f(x,y) and any α and ß

This shows that the response at any point in the image depends only on the input at that point and not on its position.

Linear, Position-Invariant Degradations

- In terms of a continuous impulse function

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha,y-\beta)d\alpha\, d\beta$$

$$g(x,y)=H[f(x,y)]$$
$$=H\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\delta(x-\alpha,y-\beta)d\alpha\, d\beta\right]$$

Linear, Position-Invariant Degradations

$$g(x,y)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} H\left[f(\alpha,\beta)\delta(x-\alpha,y-\beta)\right]d\alpha\, d\beta$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)H\left[\delta(x-\alpha,y-\beta)\right]d\alpha\, d\beta$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x,\alpha,y,\beta)d\alpha\, d\beta$$

Linear, Position-Invariant Degradations

- Impulse response of H

$$h(x,\alpha,y,\beta) = H[\delta(x-\alpha, y-\beta)]$$

- In optics, the impulse becomes a point of light
- Point spread function (PSF)

$$h(x,\alpha,y,\beta)$$

- All physical optical systems blur (spread) a point of light to some degree

Linear, Position-Invariant Degradations

- Superposition (or Fredholm) integral of the first kind

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) h(x, \alpha, y, \beta) d\alpha\, d\beta$$

- It states that if the response of H to an impulse is known, the response to any input f(α,ß) can be calculated
- So it can be said that a linear system H is completely characterized by its impulse response.

Linear, Position-Invariant Degradations

- If H is position invariant

$$H[\delta(x-\alpha, y-\beta)] = h(x-\alpha, y-\beta)$$

- Convolution integral

$$g(x, y) =$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha, \beta)h(x-\alpha, y-\beta)d\alpha\, d\beta$$

Linear, Position-Invariant Degradations

- In the presence of additive noise

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x,\alpha,y,\beta)d\alpha\, d\beta + \eta(x,y)$$

  - If H is position invariant

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)h(x-\alpha,y-\beta)d\alpha\, d\beta + \eta(x,y)$$

Linear, Position-Invariant Degradations

- If H is position invariant

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

- Restoration approach
  - Image deconvolution
  - Deconvolution filter

Estimating the Degradation Function

- Estimation by image observation
  - In order to reduce the effect of noise in our observation, we would look for areas of strong signal content

$$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

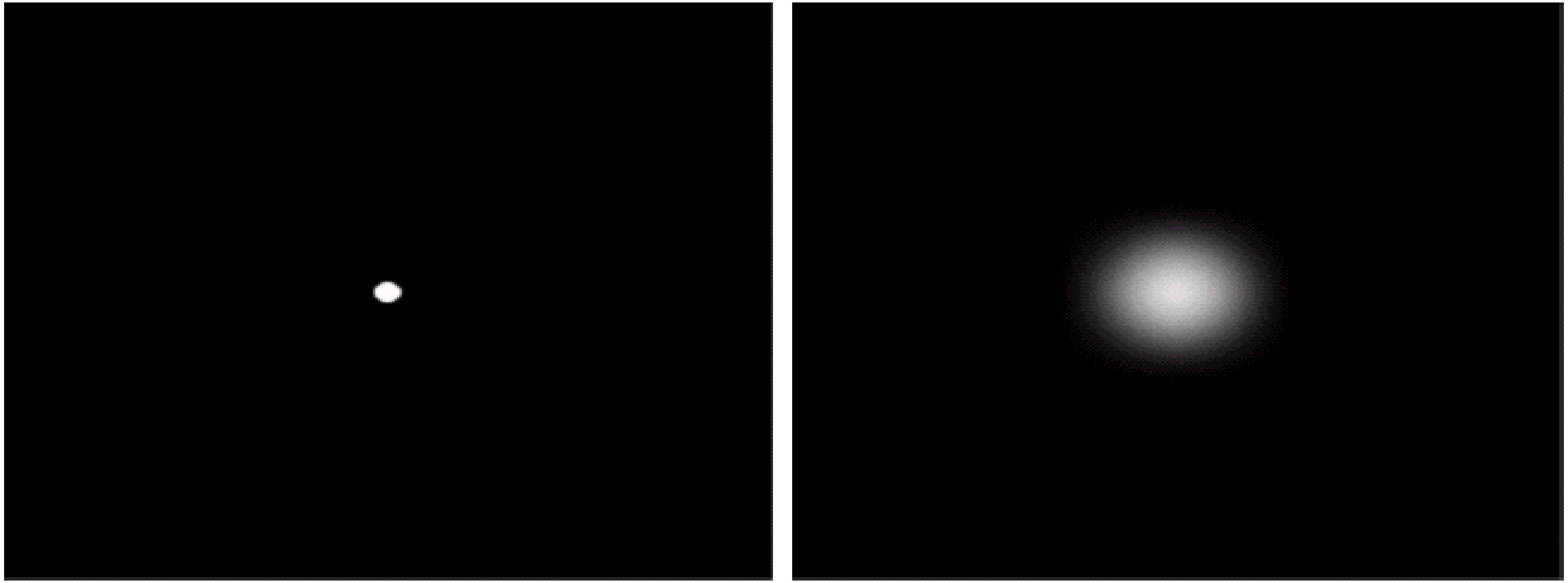
Estimating the Degradation Function

- Estimation by experimentation
  - Obtain the impulse response of the degradation by imaging an impulse (small dot of light) using the same system settings

$$H(u,v) = \frac{G(u,v)}{A}$$

  - Observed image $G(u,v)$
  - The strength of the impulse A

Estimating the Degradation Function



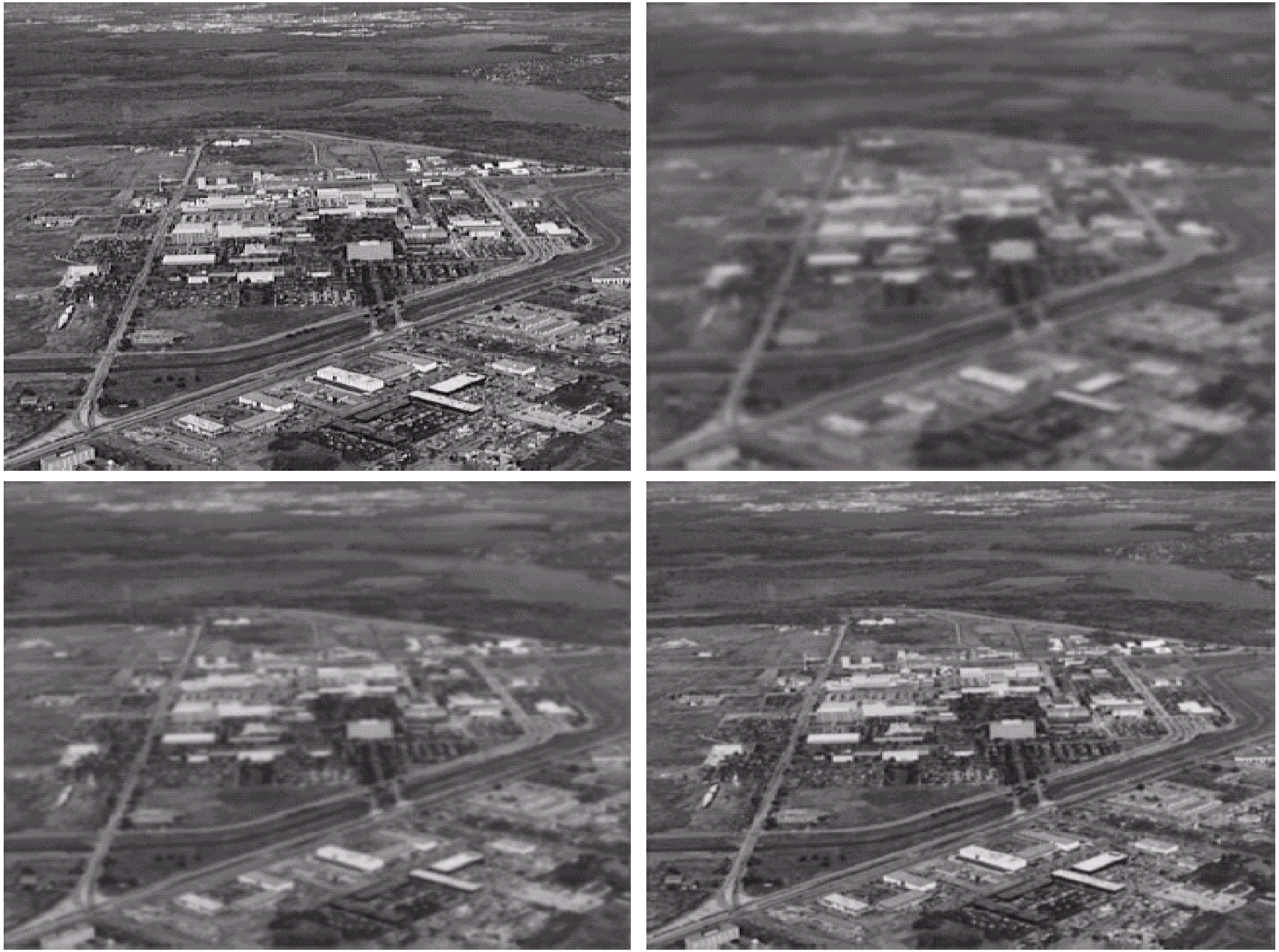
a b

FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

Estimating the Degradation Function

- Estimation by modeling
  - Hufnagel and Stanley
  - Physical characteristic of atmospheric turbulence

$$H(u,v) = e^{-k(u^2+v^2)^{\frac{5}{6}}}$$



a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

Estimating the Degradation Function

- Image motion

$$g(x,y)=\int_0^T f[x-x_0(t),y-y_0(t)]dt$$

Estimating the Degradation Function

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux+vy)}dx\,dy$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_{0}^{T} f[x-x_0(t), y-y_0(t)]dt\right]$$

$$e^{-j2\pi(ux+vy)}dx\,dy$$

$$= \int_{0}^{T}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x-x_0(t), y-y_0(t)]\right.$$

$$\left. e^{-j2\pi(ux+vy)}dx\,dy\right]dt$$

Estimating the Degradation Function

$$G(u,v) = \int_0^T F(u,v) e^{-j2\pi[u\,x_0(t)+v\,y_0(t)]} dt$$

$$= F(u,v) \int_0^T e^{-j2\pi[u\,x_0(t)+v\,y_0(t)]} dt$$

$$= F(u,v) H(u,v)$$

- Where

$$H(u,v) = \int_0^T e^{-j2\pi[u\,x_0(t)+v\,y_0(t)]} dt$$

Degradation due to Linear Motion

- If $x_0(t) = at/T$ and $y_0(t) = 0$

$$H(u,v) = \int_0^T e^{-j2\pi[u\,x_0(t)]}dt$$

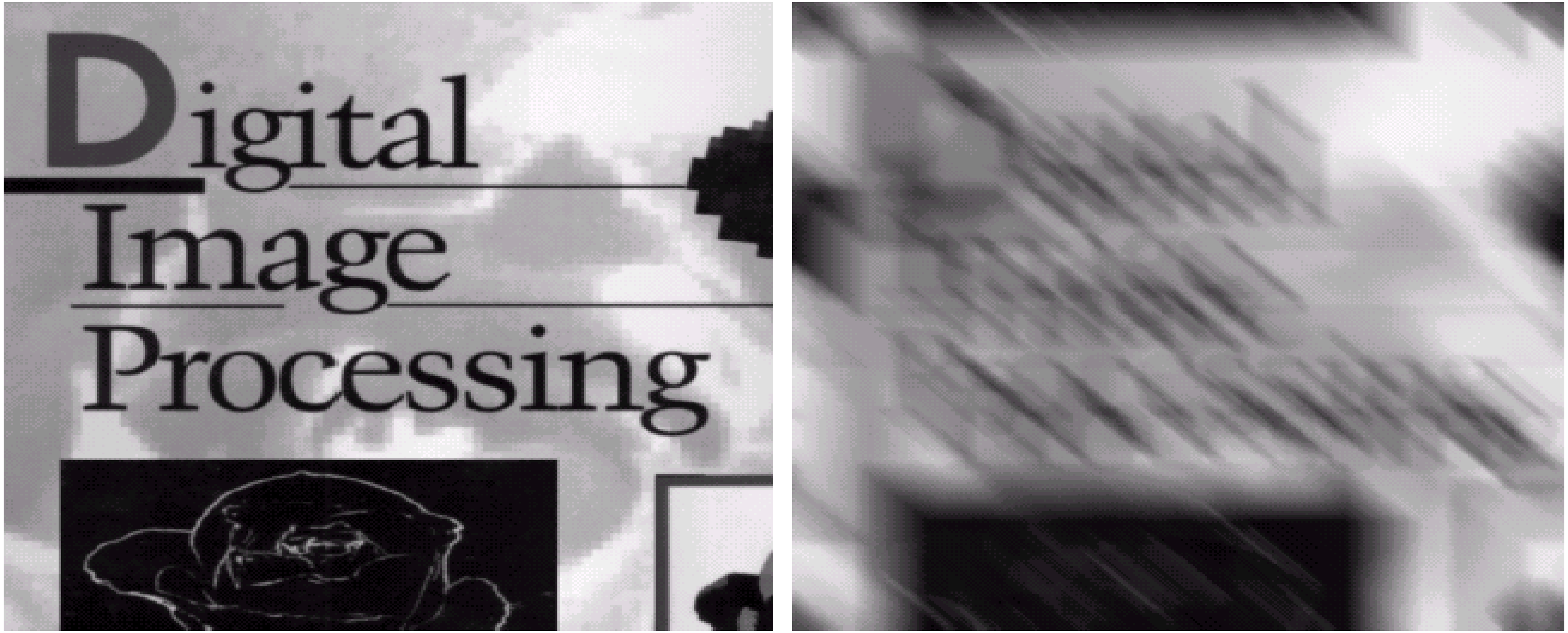
$$= \int_0^T e^{-j2\pi[u\,at/T]}dt$$

$$= \frac{T}{\pi\,ua}\sin(\pi\,ua)e^{-j\pi\,ua}$$

Degradation due to Linear Motion

- If $x_0(t) = at/T$ and $y_0(t) = bt/T$

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin[\pi(ua+vb)] e^{-j\pi(ua+vb)}$$



a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Inverse Filtering

- Direct inverse filtering: Simplest approach to restoration where an estimate $\hat{F}(u,v)$ of the transform is computed as follows. $G(u,v)$ is substituted for in the next line :

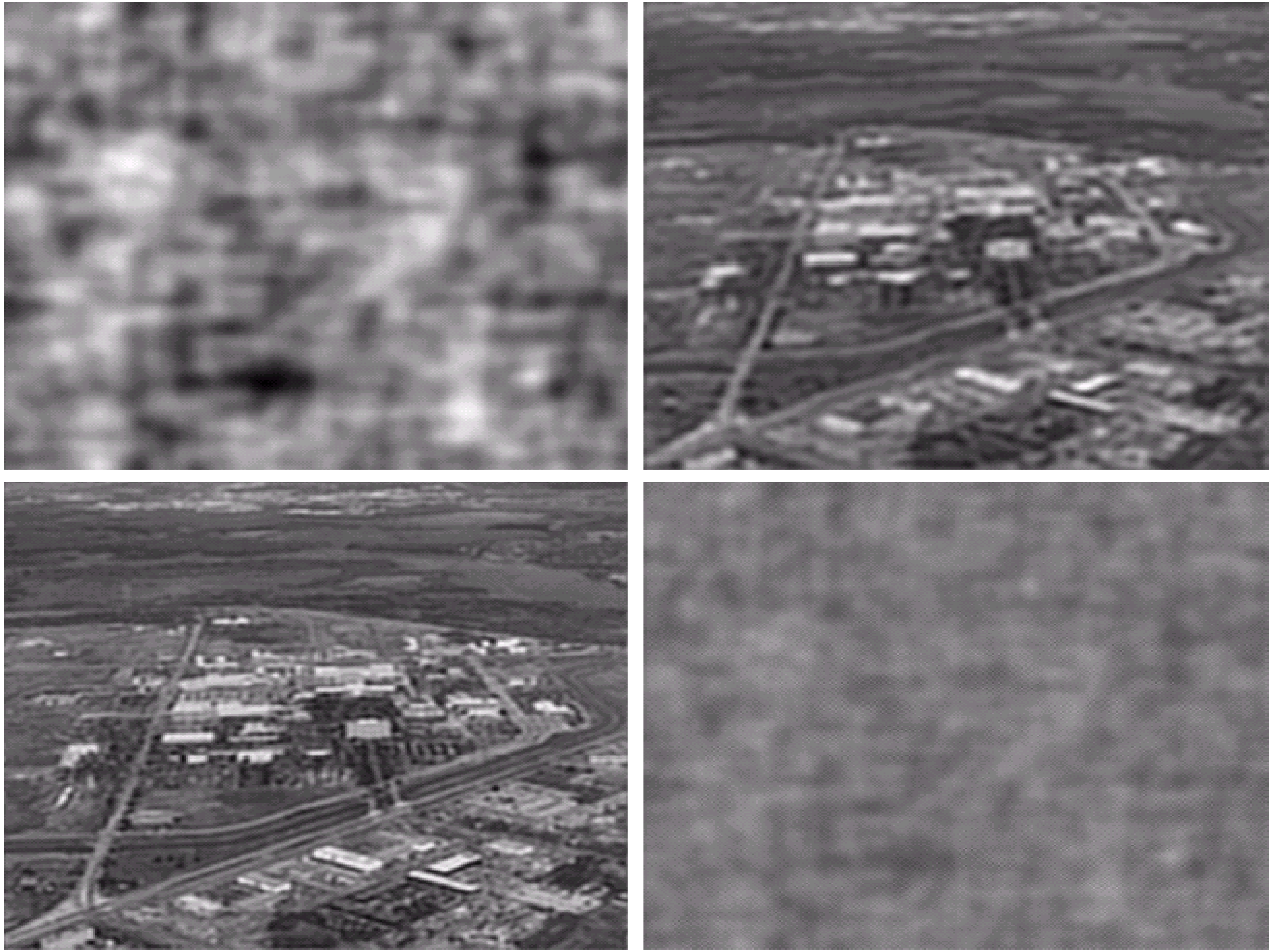
$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

$$= F(u,v) + \frac{N(u,v)}{H(u,v)}$$

- This expression tells us that even if the degradation function is known the undegraded image cannot be recovered[the inverse Fourier transform of F(u,v)] as N(u,v) is unknown.
- If the degradation function has zero or very small values then the ratio N(u,v)/H(u,v) could easily dominate the estimate $\hat{F}(u,v)$
- This can be prevented by limiting the filter frequencies to values near the origin as H(0,0) is usually the highest value of H(u,v) in frequency domain.

a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.



Minimum Mean Square Error (Wiener) Filtering

- Minimize $e^2 = E\{(f - \hat{f})^2\}$
- Terms
  - $H(u,v)$ = degradation function
  - $H^*(u,v)$ = complex conjugate of $H(u,v)$
  - $|H(u,v)|^2 = H^*(u,v)H(u,v)$
  - $S_\eta(u,v) = |N(u,v)|^2 =$ power spectrum of the noise
  - $S_f(u,v) = |F(u,v)|^2 =$ power spectrum of the undegraded image

Wiener Filter

- Wiener filter

$$\hat{F}(u,v) = \left[\frac{H^*(u,v) S_f(u,v)}{S_f(u,v)\left|H(u,v)\right|^2 + S_\eta(u,v)} \right] G(u,v)$$

$$= \left[\frac{H^*(u,v)}{\left|H(u,v)\right|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

$$= \left[\frac{1}{H(u,v)} \frac{\left|H(u,v)\right|^2}{\left|H(u,v)\right|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v)$$

- Using the property that the product of a complex quantity with its conjugate=the magnitude of the complex quantity squared.
- The result was first proposed by N. Wiener in 1942 and the Wiener filter consists of the terms inside the bracket, also came to be known as minimum mean square error or least square error filter.

Wiener Filter

- Dealing with spectrally white noise, the spectrum $|N(u,v)|^2$ is a constant.
- However the power spectrum of the undegraded image is seldom known.
- The following expression is used when these quantities are not known.

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K} \right] G(u,v)$$

- K is a specified constant that is added to all the terms of $|H(u,v)|^2$

Inverse and Wiener Filter Results

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

Effect of Wiener Filter on Motion Blur and Noise

a b c
d e f
g h i

FIGURE 5.29 (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

Constrained Least Squares Filtering

- In the absence of the power spectra and noise of the undegraded image approximations or estimates are used in Wiener filter with good results.
- But a constant estimate of the ratio of the power spectra is not always satisfactory.
- Also the Wiener filter is based on minimizing a statistical criterion and hence gives an optimal solution in an average sense
- The Constrained Least Squares Filter yields an optimal solution for each image to which it is applied.
- Vector-matrix form $g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta}$$

- $\mathbf{g}, \ \mathbf{f}, \ \boldsymbol{\eta} : MN \times 1$
- $\mathbf{H} : MN \times MN$
- Therefore the restoration problem reduces to simple matrix manipulations.

Constrained Least Squares Filtering

- Restoration is constrained by the parameters of the problems
- Minimize

$$C = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[\nabla^2 f(x,y)\right]^2$$

  - Subject to $\left\|\mathbf{g} - \mathbf{H}\hat{\mathbf{f}}\right\|^2 = \left\|\boldsymbol{\eta}\right\|^2$
  - Where $\left\|\mathbf{w}\right\|^2 = w\mathrm{T}\mathbf{w}$ is the Euclidean vector norm and f^ is the estimate of the undegraded image.

Constrained Least Squares Filtering

- The solution to this optimization problem is

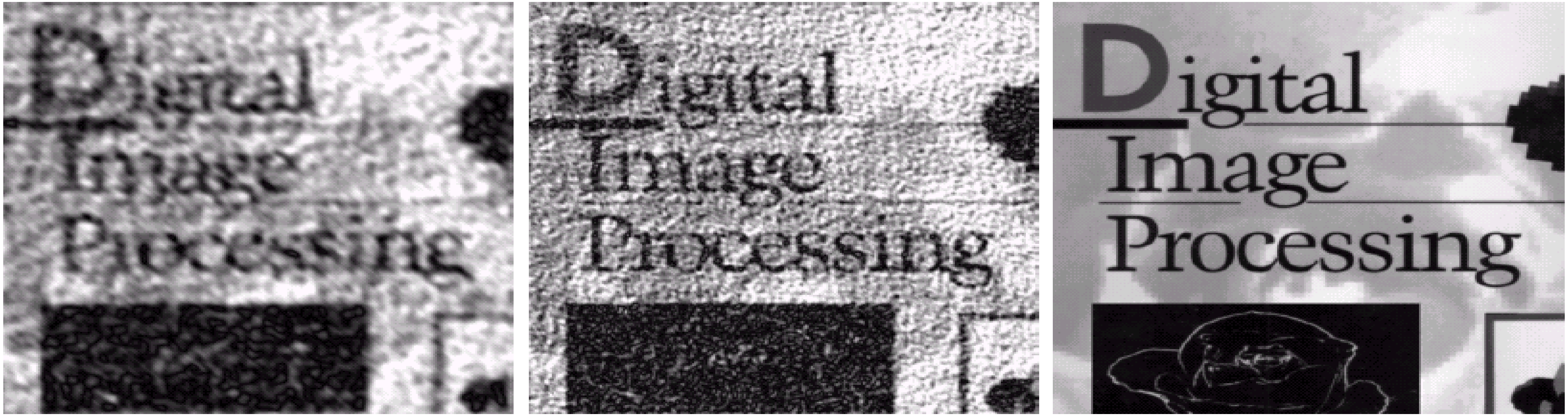
$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |P(u,v)|^2} \right] G(u,v)$$

- Where $P(u,v)$ is the Fourier transform of the function

$$P(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

- And the parameter Υ is adjusted to satisfy the above constraint. For Υ=0, this equation reduces to inverse filtering.

Constrained Least Squares Filtering



a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

Constrained Least Squares Filtering

- Computing γ by iteration (where a residual vector r is defined)

$$\mathbf{r} = \mathbf{g} - \mathbf{H}\hat{\mathbf{f}}$$

  - Adjust γ so that

$$\|\mathbf{r}\|^2 = \|\boldsymbol{\eta}\|^2 \pm a$$

  - Where a is an accuracy factor.

Constrained Least Squares Filtering

- Computation

$$\|\mathbf{r}\|^2 = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} r^2(x, y)$$

$$m_\eta = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \eta(x, y)$$

$$\sigma_\eta^2 = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} \left[\eta(x, y) - m_\eta\right]^2$$

$$\|\boldsymbol{\eta}\|^2 = MN[\sigma_\eta^2 + m_\eta^2]$$

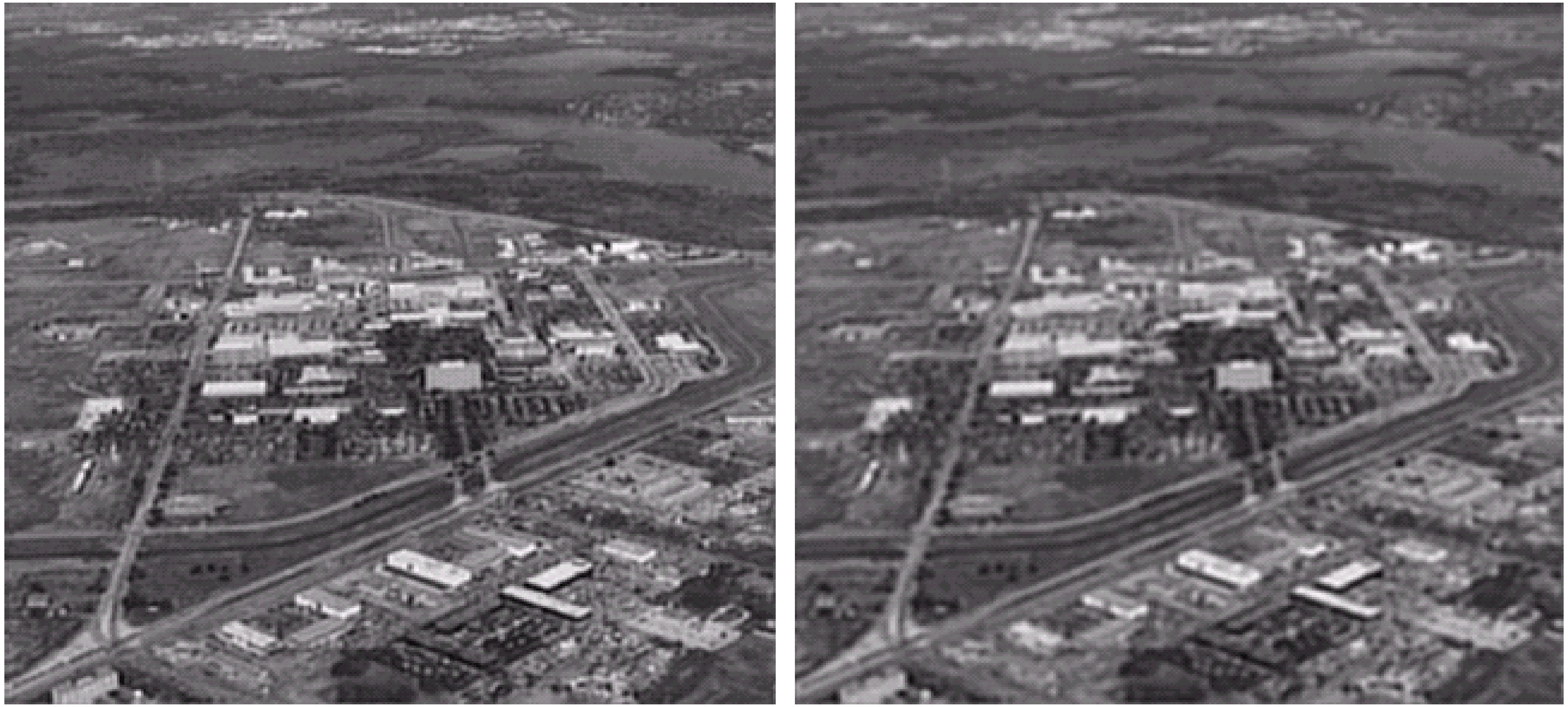
Constrained Least Squares Filtering

- Algorithm
- 1: Specify an initial value of γ
- 2: Compute $\|\mathbf{r}\|^2$
- 3: Stop if $\|\mathbf{r}\|^2 = \|\boldsymbol{\eta}\|^2 \pm a$ is satisfied; otherwise return to Step 2 after increasing γ if $\|\mathbf{r}\|^2 < \|\boldsymbol{\eta}\|^2 - a$ or
- decreasing γ if $\|\mathbf{r}\|^2 > \|\boldsymbol{\eta}\|^2 + a$.
- Use a new value of γ to recompute the optimum estimate $\hat{F}(u,v)$

Constrained Least Squares Filtering

a b

FIGURE 5.31
(a) Iteratively determined constrained least squares restoration of Fig. 5.16(b), using correct noise parameters.
(b) Result obtained with wrong noise parameters.



Geometric Mean Filter

- Generalization of the Wiener filter results in the following filter:

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2}\right]^{\alpha}\left[\frac{H^*(u,v)}{|H(u,v)|^2 + \beta\left[\frac{S_\eta(u,v)}{S_f(u,v)}\right]}\right]^{1-\alpha} G(u,v)$$

- With α and ß being positive real constants.
- When α=1 this filter reduces to inverse filter.
- When α=0 this filter reduces to parametric Wiener filter and becomes the standard Wiener filter with ß=1.
- If α=1/2, the filter becomes a product of the two quantities raised to the same power,i.e. the Geometric mean
- With α=1/2 and ß=1, the filter is commonly referred to as the Spectrum Equalization Filter

Geometric Transformations

- Almost all image capturing processes involve unwanted geometric transforms.

They are caused by

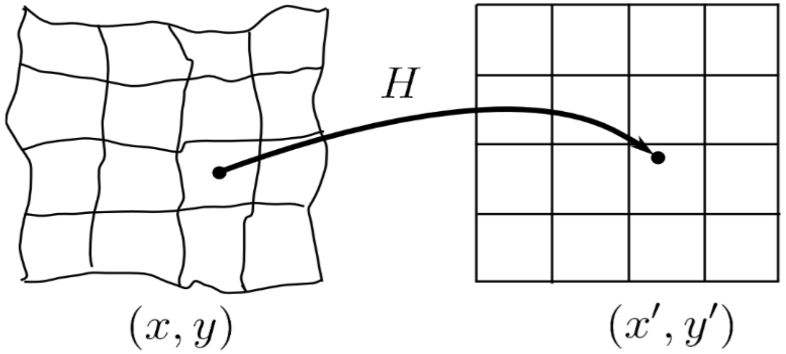
- Perspective distortions
- Optical distortions due to lens errors or aberration
- Capturing process inherent limitations , and so forth
- Geometric transforms permit to eliminate, to a large extent, these distortions.

Only after correcting these errors it would be possible to

- Derive accurate metric measurements from the images
- Compare the same or similar objects in different images

Geometric Transformations

- A geometric transform is a vector function H that maps all the pixels (x,y)in the source image to a new (x´,y´) position in the rectified coordinate system with (x´,y´) = H (x´,y´)
- Fig 5.1: Geometrical transform



- The transformation H is either known in advance or can be determined from several known pixel correspondences in an original and transformed image pair.
- Depending on the geometrical distortion one has to select the most appropriate geometrical transformation from a class of transformations.

Basic Steps of a Geometric Transform

- Such geometric transforms consist of two basic steps:

1. Determining the pixel coordinates in the transformed image
    - Mapping of the coordinates (x,y) in the input image to the point (x′,y′) in the output image
    - The output coordinates generally don't fall onto exact pixel coordinates
2. Determining the point in the digital raster which matches the transformed point and determining its brightness/colour
    - Brightness/colour is usually computed as an interpolation of several points in the neighbourhood

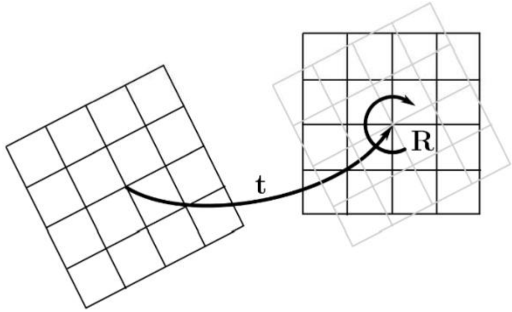
The Hierarchy of Geometric Transformations

- The projective transformations form in a mathematical sense a group known as the projective linear group. This group can be split into various sub-groups with special properties and increasing complexity (degrees-of-freedom DOF).
  - Class I: Isometries (translation, rotation, rigid) Class II: Similarity transformations
  - Class III: Affine transformations Class IV: Projective transformations
- The most general case of geometric transformation is the Free-form or curved transformation
- It, however, doesn't fit in the nice mathematical framework of the projective linear group.

Class I: Isometries

- Isometries are transformations of the plane that preserve Euclidean distance thus the term isometry (from iso = same, metric = measure).

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon\cos\theta & -\sin\theta & t_x \\ \epsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



- Fig 5.3: Isometry =
- Translation + Rotation

- Fig.(5.1)
- where $\epsilon = \pm 1$ If $\epsilon = 1$ then the isometry is orientation- preserving and is composed of a translation and rotation. If $\epsilon = -1$ then the isometry reverses orientation.
- The above equation can be written more concisely in block form as $\mathbf{x}' = H_E\mathbf{x} = \begin{bmatrix} \mathrm{R} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{bmatrix} \mathbf{x}$ Fig. (5.2)
- where **R** is a 2x2 (orthogonal) rotation matrix, **t**, a translation vector, and **0** a null vector.

Class I: Isometries

- Degrees of Freedom
  - Isometries have 3 degrees-of-freedom (DOF): one for rotation (Ø) and two for translation(tx, ty), The Isometry can be computed from two point correspondences

- Invariants
  - Lengths (distance between two points) Angles (angle between two lines)
  - Area

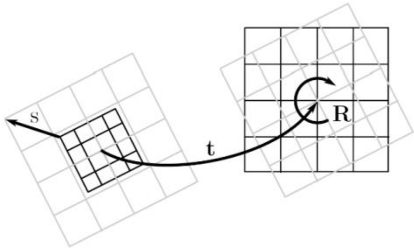
Class II: Similarity Transformations

- A similarity transformation is an isometry composed with an isotropic scaling, i.e.

Fig. (5.4)

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Fig 5.6: Similarity =
- Translation + Rotation + Isotropic Scaling



- The above equation can be written more concisely in block form as

$$\mathbf{x}' = H_S\mathbf{x} = \begin{bmatrix} s\mathrm{R} & \mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{bmatrix} \mathbf{x}$$

- (5.5)

- where **R** is a 2x2 (orthogonal) rotation matrix, **t**, a translation vector, and **0** a null vector

Class II: Similarity Transformations (2)

Degrees of Freedom

- Similarity Transformations have 4 DOF: one for rotation, two for translation , and one for scaling
- A similarity can be computed from two point correspondences

Invariants

- Angles (angle between two lines)
- Parallel lines are mapped to parallel lines
- The length between two points is not invariant, but the ratio of two lengths is
- Similarly the ratio of areas is invariant (because the squared scaling cancels out)

Class II: Similarity Transformations Example

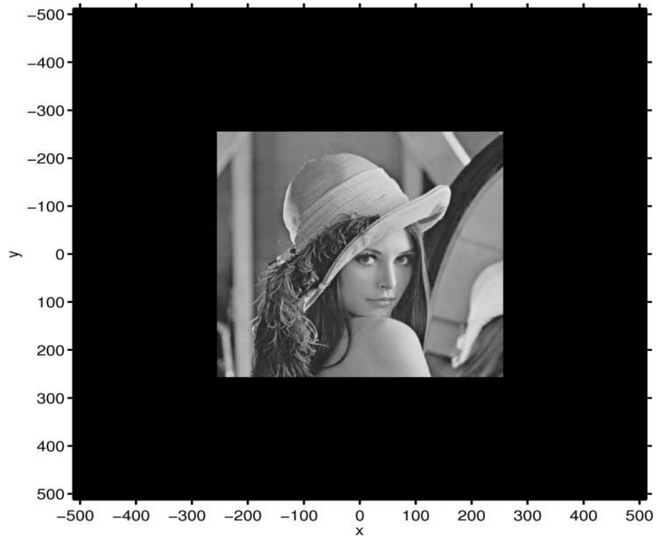


Fig 5.7: Original image

Similarity transformation with

$$\theta = 45^\circ$$

$$\mathbf{t} = \begin{bmatrix} -100 \\ -100 \end{bmatrix}^T$$

$$s = 0.5$$

yielding a transformation matrix

$$H = \begin{bmatrix} .3536 & -.3536 & -100 \\ .3536 & .3536 & -100 \\ 0 & 0 & 1 \end{bmatrix}$$

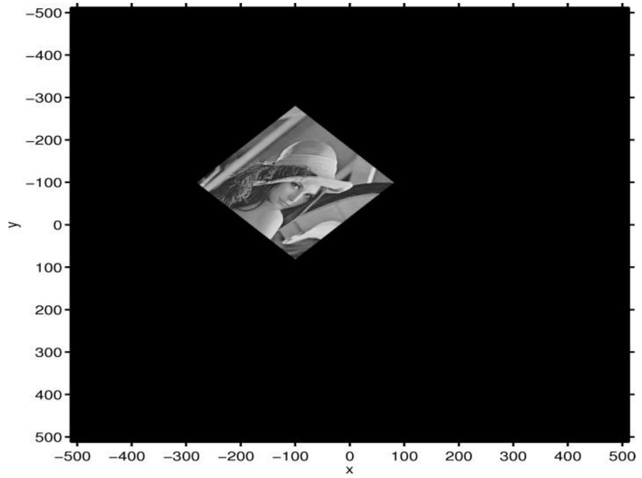


Fig 5.8: Image after similarity transformation

Class II: Practical Issues

- The matrix elements of H should not be estimated directly

$$H = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Estimate the *Euler angle* θ *scaling s, and translation parameters* t_x, t_y *instead*.

Class III: Affine Transformations

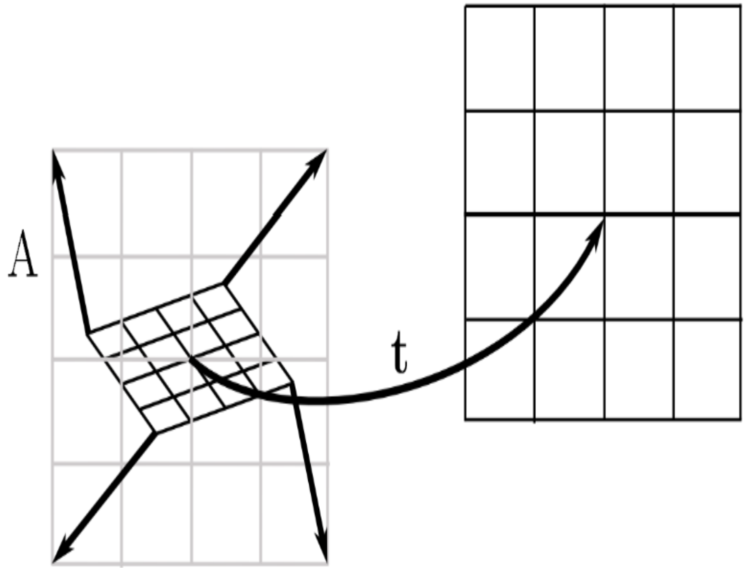
An affine transformation is a non-singular linear transformation (rotation, scaling and skewing) followed by a translation, i.e.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

The above equation can be written more concisely in block form as

$$\mathbf{x}' = H_A \mathbf{x} = \begin{bmatrix} \mathrm{A} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \mathbf{x} \qquad (5.8)$$

Where A is a 2x2 non-singular matrix, t a translation vector and 0a null vector.



fFig 5.9: Affine = Non-singular linear transformation + Translation

An affinity is orientation-preserving or -reversing if $\det(\mathrm{A})$ Is positive or negative respectively .

Class III: Affine Transformations

Degrees of Freedom

Affine Transformations have 6 DOF: four for the non-singular matrix **A**, and two for translation t_x, t_y

An affinity can be computed from three point correspondences

Invariants

Parallel lines are mapped to parallel lines Ratio of lengths of parallel line segments Ratio of areas

Class III: Affine Transformation

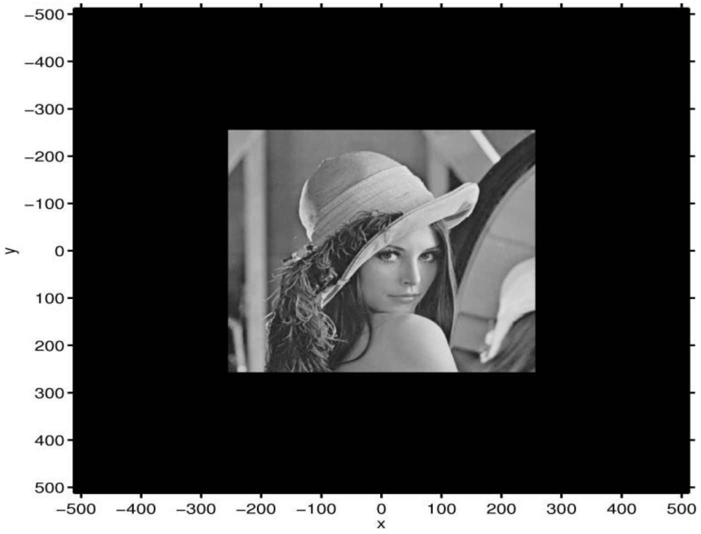


Fig 5.10: Original image

Affine transformation with

$$\mathbf{t} = \begin{bmatrix} -100 \\ -100 \end{bmatrix}^T$$

$$A = \begin{bmatrix} 0.5305 & -0.2652 \\ 0.5305 & 0.5305 \end{bmatrix}$$

yielding a transformation matrix

$$H = \begin{bmatrix} 0.5305 & -0.2652 & -100 \\ 0.5305 & 0.5305 & -100 \\ 0 & 0 & 1 \end{bmatrix}$$

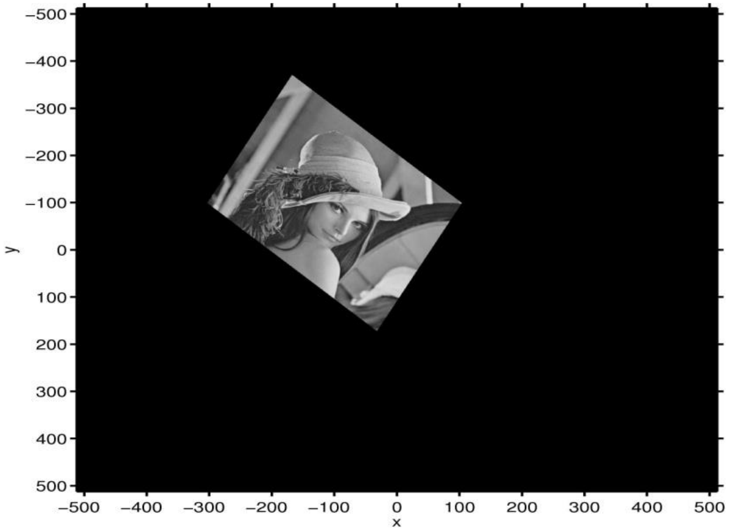


Fig 5.11: Image after affine transformation

Class III: Practical Issues

- The use of the *affine transform* rather than the similarity transform (rigid) *does not* greatly *increase* its *applicability in medicine*, as there are *no organs* that *only stretch or shear*.
- Tissues normally deform in more complex non-rigid ways.

- There are, however, scanner introduced errors than can result in skewing terms

  - *Tilted gantry* in CT acquisitions
  - *Tilted plane* in optical image acquisitions such as microscope or endoscope

  and affine transforms are used to overcome these problems.

  - Nevertheless, the *affine transformation* is *often used* for registration, as *no care has to be taken* that $a_{11}, a_{12}, a_{21}, a_{22}$ *form a proper rotation matrix*.

Class IV: Projective Transformations

A projective transformation is a general non-singular linear transformation of homogeneous coordinates, i.e.

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad (5.9)$$

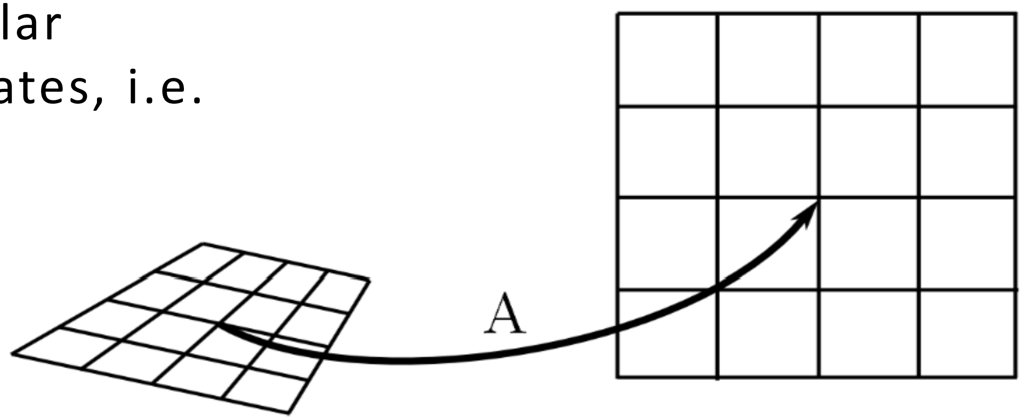


Fig. 5.12: General non-singular linear transformation

The above equation is often written more concisely in block form as

$$\mathbf{x}' = H\mathbf{x} = \begin{bmatrix} A & \mathbf{t} \\ \mathbf{v}^\mathsf{T} & v \end{bmatrix} \mathbf{x} \qquad (5.10)$$

where $\mathbf{v}^\mathsf{T}$ is a vector $\mathbf{v}^\mathsf{T} = (v_1, v_2)$, A a non-singular matrix, T a translation vector, and v the scaling parameters.

Class IV: Projective Transformations

- Degrees of Freedom
  - Projective Transformations have 8 DOF
  - A projective transformation can be computed from *four point* correspondences (with no three collinear on either plane)
- Invariants
  - The most fundamental projective invariant is the cross ratio (ratio of ratio) of four collinear points

Class IV: Projective Transformation Example

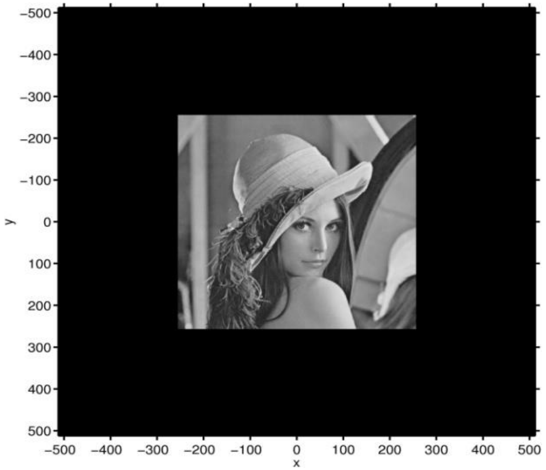


Fig 5.13: Original image

Projective transformation with a transformation matrix

$$H = \begin{bmatrix} .5305 & -.2652 & -100 \\ .5305 & .5305 & -100 \\ 0.001 & 0 & 1 \end{bmatrix}$$

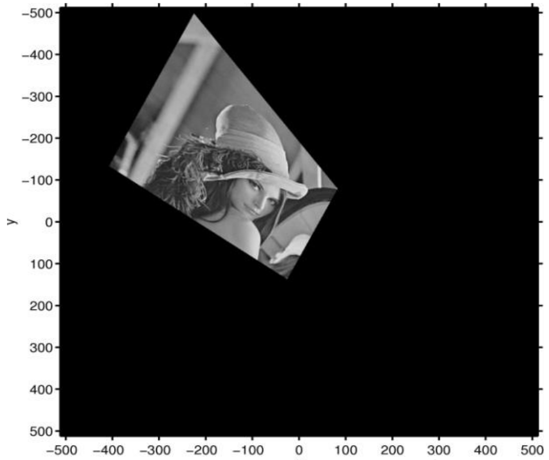


Fig 5.14: Image after projective transformation

Geometric Transformations

- Spatial transformations

$$x' = r(x, y)$$

- Tiepoints

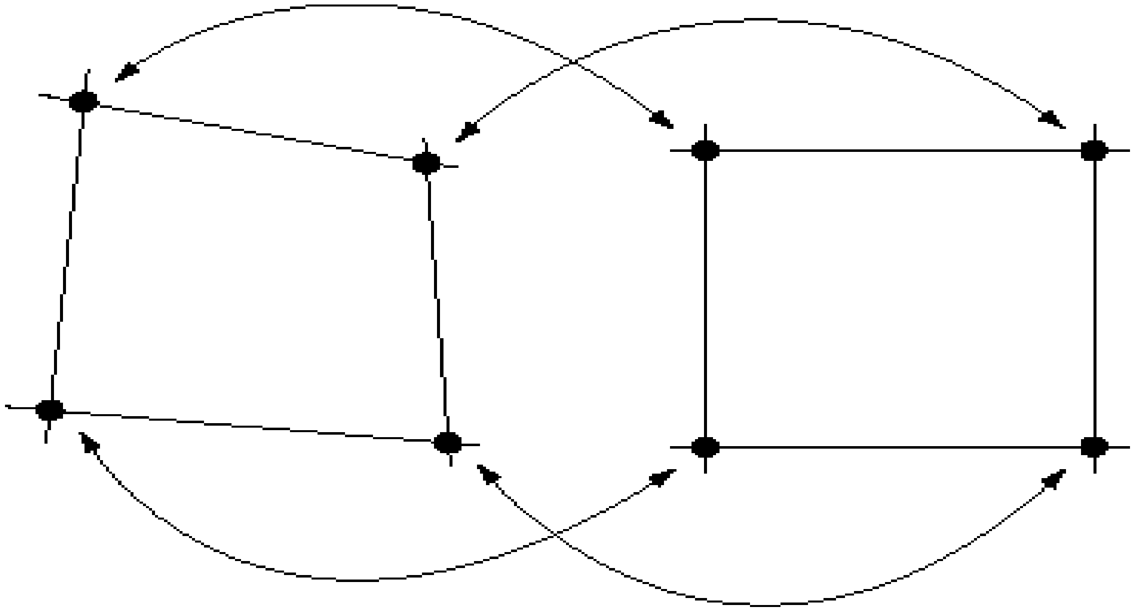


FIGURE 5.32 Corresponding tiepoints in two image segments.

Geometric Transformations

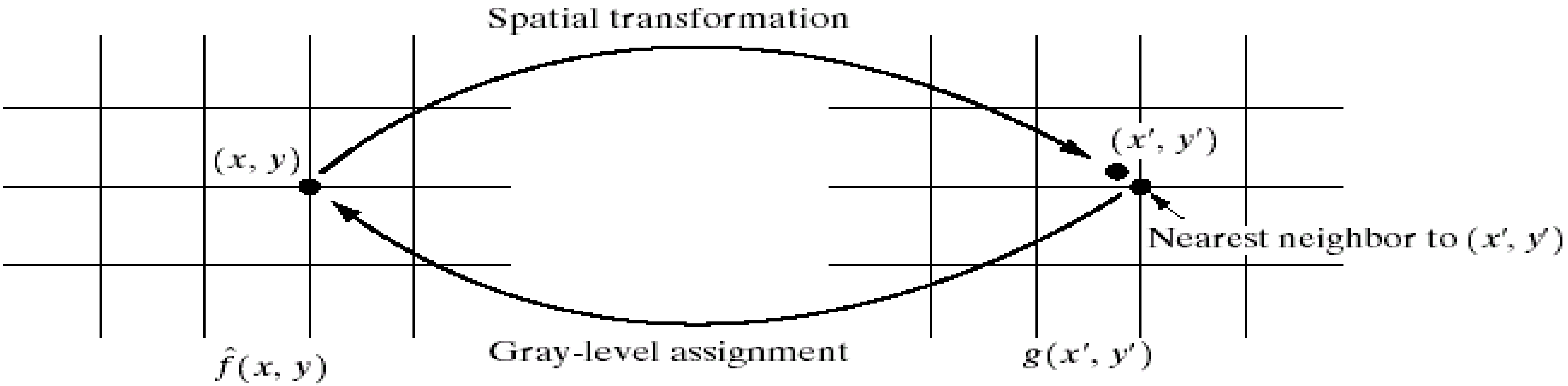
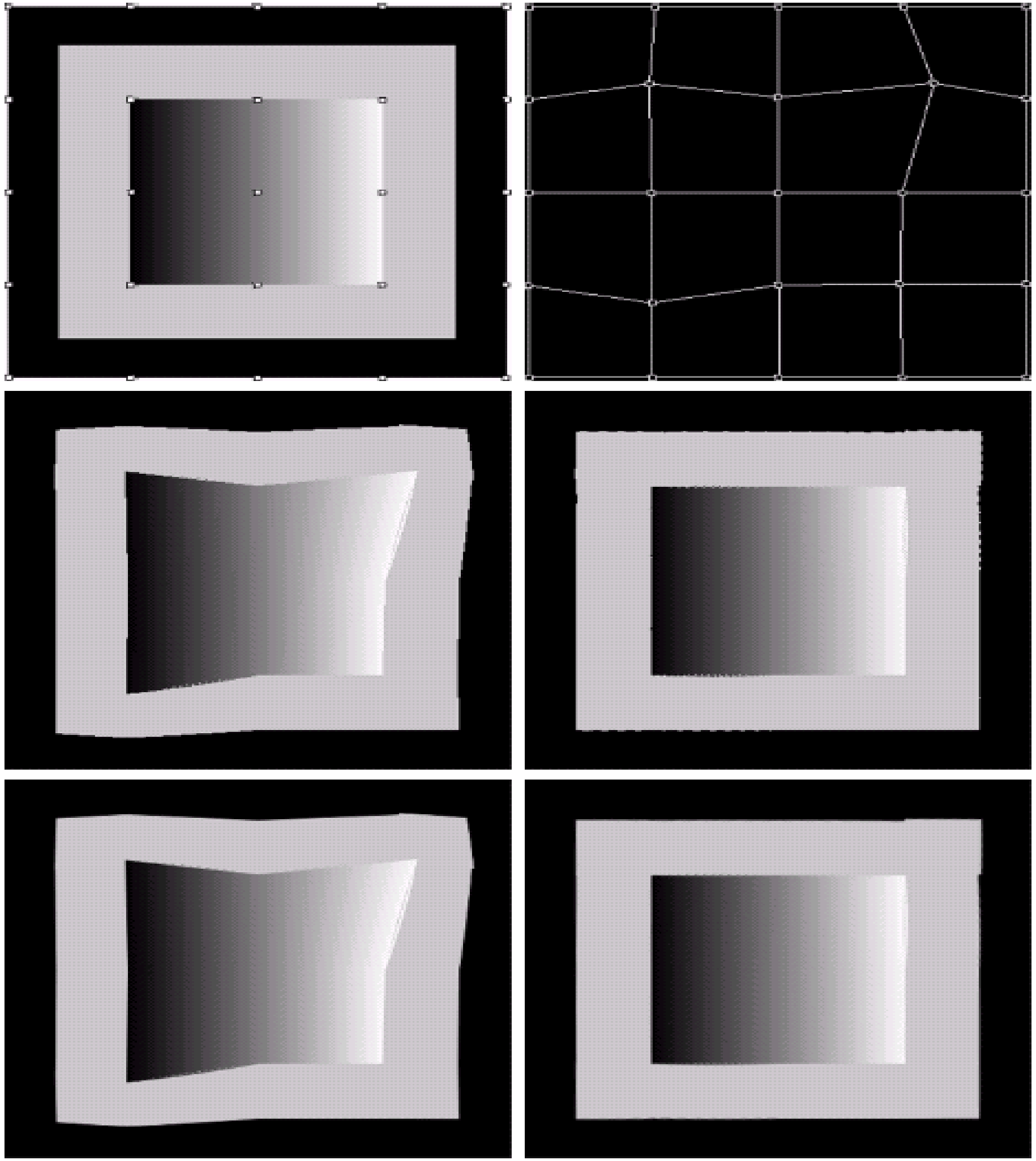


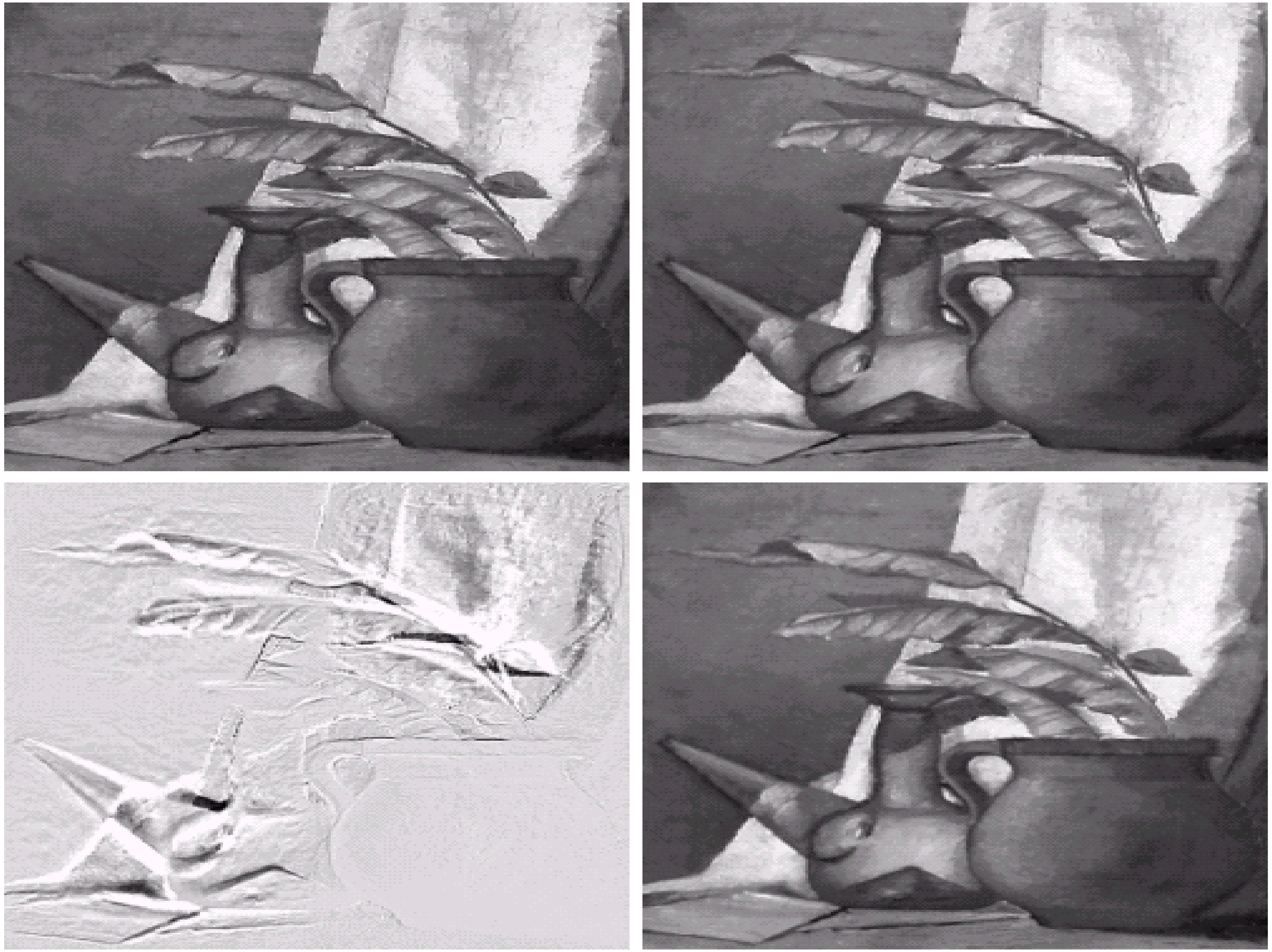
FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.

Geometric Transformations



a b
c d
e f

FIGURE 5.34 (a) Image showing tiepoints. (b) Tiepoints after geometric distortion. (c) Geometrically distorted image, using nearest neighbor interpolation. (d) Restored result. (e) Image distorted using bilinear interpolation. (f) Restored image.



a b
c d

FIGURE 5.35 (a) An image before geometric distortion. (b) Image geometrically distorted using the same parameters as in Fig. 5.34(e). (c) Difference between (a) and (b). (d) Geometrically restored image.

Chapter 5

END OF LECTURE